B.TECH. COMPUTER SCIENCE AND ENGINEERING

COURSE STRUCTURE

Approved in Academic Council held on 28 July 2021

B.TECH. COMPUTER SCIENCE AND ENGINEERING

PROGRAM OBJECTIVES

PO 1 Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and engg. specialization to the solution of complex engineering problems.

PO-2 Problem analysis: Identify, formulate, research literature, and analyze engineering problems to arrive at substantiated conclusions using first principles of mathematics, natural, and engineering sciences.

PO-3 Design/development of solutions: Design solutions for complex engineering problems and design system components, processes to meet the specifications with consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO-4 Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO-5 Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO-6 The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO-7 Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO-8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO-9 Individual and team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

PO-10 Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

PO-11 Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one’s own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

PO-12 Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN

COURSE CURRICULUM OF CSE&IT DEPARTMENT- 2021 Batch (160 CREDITS)

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 1st SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	HSS	21B11HS111	English	2	0	0	**2**	**2**
2	Basic Sciences	18B11MA111	Engineering Mathematics -1	3	1	0	**4**	**4**
3	Basic Sciences	18B11PH111	Engineering Physics-I	3	1	0	**4**	**4**
4	Engg Science	19B11CI111	Programming for Problem Solving-II	2	0	0	**2**	**2**
5	Engg Science	18B17GE171	Workshop Practices **OR**	0	0	3	**1.5**	**3**
6		18B17GE173	Engineering Graphics	0	0	3		**3**
7	Basic Sciences	18B17PH171	Engineering Physics Lab-I	0	0	2	**1**	**2**
8	Engg Science	19B17CI171	Programming for Problem Solving Lab-II	0	0	4	**2**	**4**
9	HSS	21B17HS171	English Lab	0	0	2	**1**	**2**
10		18B17GE172	Mandatory Induction Program	-	-	-	-	-
						Total	**17.5**	**23**

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 2nd SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Basic Sciences	18B11MA211	Engineering Mathematics -II	3	1	0	**4**	**4**
2	Basic Sciences	18B11PH211	Engineering Physics-II	3	0	0	**3**	**3**
3	Basic Sciences	18B11PH271	Engineering Physics Lab - II	0	0	2	**1**	**2**
4	Engg Science	18B11EC211	Electrical Sciences	3	1	0	**4**	**4**
5	Engg Science	18B17EC271	Electrical Sciences Lab	0	0	2	**1**	**2**
6	Engg Science	18B17GE173	Engineering Graphics **OR**	0	0	3	**1.5**	**3**
7		18B17GE171	Workshop Practices	0	0	3		**3**
8	Engg Science	18B11CI211	Data Structures and Algorithms	3	1	0	4	4
9	Engg Science	18B17CI271	Data Structures and Algorithms Lab	0	0	4	**2**	**4**
10	HSS	21B11HS211	Life Skills & Effective Communication	1	0	0	**1**	**1**
11	HSS	21B17HS271	Life Skills & Effective Communication	0	0	2	**1**	**2**
						Total	**22.5**	**32**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN

COURSE CURRICULUM OF CSE&IT DEPARTMENT- 2021 Batch (160 CREDITS)

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 3rd SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Core	18B11CI314	Python Programming Essentials	3	0	0	**3**	**3**
2	Engg Science	18B11CI311	Object Oriented Systems and Programming	3	0	0	**3**	**3**
3	Professional Core	18B11CI313	Database Management systems	3	0	0	**3**	**3**
4	Basic Sciences	18B11MA313	Probability & Statistics	3	0	0	**3**	**3**
5	HSS	21B11HS312	Interpersonal Dynamics, Values and Ethics	1	0	0	**1**	**1**
6	Professional Core	18B17CI374	Python programming Lab	0	0	4	**2**	**4**
7	Engg Science	18B17CI371	Object Oriented Systems and Programming Lab	0	0	4	**2**	**4**
8	Professional Core	18B17CI373	Database Management Systems Lab	0	0	4	**2**	**4**
9	Engg Science	18B17CI372	IT Workshop (SciLab/MATLAB) Lab	0	0	4	**2**	**4**
10	HSS	21B11HS311	Professional Communication Practice	0	0	2	**Audit**	**2**
						Total	**21**	**31**

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 4th SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Core	18B11CI414	Discrete Computational Mathematics	3	0	0	**3**	**3**
2	Professional Core	18B11CI413	Modeling and Simulation Techniques	2	0	0	**2**	**2**
3	Professional Core	18B11CI411	Operating Systems	3	0	0	**3**	**3**
4	Professional Core	18B11CI412	Design & Analysis of Algorithms	3	0	0	**3**	**3**
5	Mandatory Course	18B11GE411	Environmental Studies	2	-	-	**-**	**2**
6	HSS	18B11HS411	Finance & Accounts	3	0	0	**3**	**3**
7	Professional Core	18B17CI473	Data Simulation Lab	0	0	4	**2**	**4**
8	Professional Core	18B17CI471	Operating System Lab	0	0	4	**2**	**4**
9	Professional Core	18B17CI472	Design and Analysis of Algorithms Lab	0	0	4	**2**	**4**
10	Professional Core	18B17CI474	Web Tech Lab	0	0	4	2	4
						Total	**22**	**32**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN

COURSE CURRICULUM OF CSE&IT DEPARTMENT- 2021 Batch (160 CREDITS)

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 5th SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Core	18B11CI515	Computer Graphics	3	0	0	**3**	**3**
2	Basic Sciences		Science Elective	3	0	0	**3**	**3**
3	Professional Core	18B11CI513	Formal Language & Automata Theory	3	0	0	**3**	**3**
4	Professional Core	18B11CI514	Computer Organization and Architecture	3	0	0	**3**	**3**
5	HSS	18B11HS511	Project Management and Entrepreneurship	3	0	0	**3**	**3**
6	Professional Core	18B17CI575	Computer Graphics Lab	0	0	4	**2**	**4**
7	Professional Core	18B17CI574	Computer Organization and Architecture Lab	0	0	2	**1**	**2**
8	Engg Science	18B1WCI575	Multimedia Lab	0	0	2	**1**	**2**
9	Professional Elective		Elective -I	2	0	0	**2**	**2**
10	Professional Elective		Elective -I Lab	0	0	2	**1**	**2**
						Total	**22**	**27**

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 6 th SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Core	18B11CI612	Compiler Design	3	0	0	**3**	**3**
2	Professional Core	18B11CI611	Computer Networks	3	0	0	**3**	**3**
3	Professional Core	18B17CI672	Compiler Design Lab	0	0	4	**2**	**4**
4	Professional Core	18B17CI671	Computer Networks lab	0	0	4	**2**	**4**
5	Professional Elective		Elective II	2	0	0	**2**	**2**
6	Professional Elective		Elective - II Lab	0	0	2	**1**	**2**
7	Professional Elective		Elective III	2	0	0	**2**	**2**
8	Professional Elective		Elective - III Lab	0	0	2	**1**	**2**
9	Open Elective		Open Elective -I (Humanities)	3	0	0	**3**	**3**
10	Project	18B19CI691	Minor Project	0	0	6	**3**	**6**
						Total	**22**	**31**

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN

COURSE CURRICULUM OF CSE&IT DEPARTMENT- 2021 Batch (160 CREDITS)

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 7 th SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective		Elective IV	2	0	0	**2**	**2**
2	Professional Elective		Elective IV Lab	0	0	2	**1**	**2**
3	Professional Elective		Elective V	3	0	0	**3**	**3**
4	Open Elective		Open Elective II / MOOC Course*	3	0	0	**3**	**3**
5	Open Elective		Open Elective III / MOOC Course*	3	1	0	**3**	**3**
6	Mandatory Course	18B11HS711	Indian Constitution	1	-	-	**-**	**1**
7	Project	18B19CI791	Major Project - I	0	0	12	**6**	**12**
						Total	**18**	**26**

B. TECH (COMPUTER SCIENCE AND ENGINEERING) 8 th SEMESTER

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective		Elective VI	3	0	0	**3**	**3**
2	Open Elective		Open Elective IV / MOOC Course*	3	0	0	**3**	**3**
3	Open Elective		Open Elective V / MOOC Course*	3	0	0	**3**	**3**
4	Project	18B19CI891	Major Project - II	0	0	12	**6**	**12**
						Total	**15**	**21**
			TOTAL CREDITS				**160**	**158.5**
			TOTAL HOURS				**216**	
			HSS				**12**	
			Basic Science				**23**	
			Engg Science				**24.5**	
			Professional Core				**51**	
			Professional Elective				**18**	
			OE				**15**	
			PROJECT				**15**	

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN

COURSE CURRICULUM OF CSE&IT DEPARTMENT- 2021 Batch (160 CREDITS)

B. TECH (COMPUTER SCIENCE AND ENGINEERING)

ELECTIVE-I

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI532	Data Compression	2	0	0	2	2
2	Professional Elective	18B1WCI533	Principal of Programming Languages	2	0	0	2	2
3	Professional Elective	18B1WCI534	Java Programming	2	0	0	2	2
4	Professional Elective	18B1WCI531	Information Theory & Coding	2	0	0	2	2
5	Professional Elective	18B1WCI572	Data Compression Lab	0	0	2	1	2
6	Professional Elective	18B1WCI573	Principal of Programming Languages Lab	0	0	2	1	2
7	Professional Elective	18B1WCI574	Java Programming Lab	0	0	2	1	2
8	Professional Elective	18B1WCI571	Information Theory & Coding Lab	0	0	2	1	2
9	Professional Elective	20B1WCI531	Foundation for Data Science and Visualization	2	0	0	2	2
10	Professional Elective	20B1WCI571	Data Science and Visualization Lab	0	0	2	1	2
11	Professional Elective	19B1WCI531	Big Data using Hadoop	2	0	0	2	2
12	Professional Elective	19B1WCI571	Big Data using Hadoop Lab	0	0	2	1	2
13	Professional Elective	19B1WCI532	Image Analysis and Pattern Recognition	2	0	0	2	2
14	Professional Elective	19B1WCI572	Image Analysis and Pattern Recognition Lab	0	0	2	1	2
15	Professional Elective	20B1WCI532	Cloud Computing: Concepts, Technology & Architecture	2	0	0	2	2
16	Professional Elective	20B1WCI572	Cloud Computing: Concepts, Technology & Architecture Lab	0	0	2	1	2
17	Professional Elective	19B1WCI533	Human-Computer Interaction	2	0	0	2	2
18	Professional Elective	19B1WCI573	Human-Computer Interaction Lab	0	0	2	1	2
19	Professional Elective	19B1WCI534	Social Media	2	0	0	2	2
20	Professional Elective	19B1WCI574	Social Media Lab (Node XL)	0	0	2	1	2

ELECTIVE-II

S.No.		Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI633	Software Testing Fundamentals	2	0	0	2	2
2	Professional Elective	18B1WCI634	Machine Learning	2	0	0	2	2
3	Professional Elective	18B1WCI637	C# and VB.NET	2	0	0	2	2
4	Professional Elective	18B1WCI631	Data Structure and Software Design	2	0	0	2	2
5	Professional Elective	18B1WCI673	Software Testing Fundamentals Lab	0	0	2	1	2
6	Professional Elective	18B1WCI674	Machine Learning Lab	0	0	2	1	2
7	Professional Elective	18B1WCI677	C# and VB.NET Lab	0	0	2	1	2
8	Professional Elective	18B1WCI671	Data Structure and Software Design Lab	0	0	2	1	2
9	Professional Elective	19B1WCI631	Digital Forensics	2	0	0	2	2
10	Professional Elective	19B1WCI671	Digital Forensics lab	0	0	2	1	2
11	Professional Elective	19B1WCI633	Computer Animation	2	0	0	2	2
12	Professional Elective	19B1WCI673	Computer Animation Lab	0	0	2	1	2
13	Professional Elective	19B1WCI634	Computer and Robot Vision	2	0	0	2	2
14	Professional Elective	19B1WCI674	Computer and Robot Vision lab	0	0	2	1	2
15	Professional Elective	19B1WCI636	Computability, Complexity & Algorithms	2	0	0	2	2
16	Professional Elective	19B1WCI676	Computability, Complexity & Algorithms Lab	0	0	2	1	2
17	Professional Elective	19B1WCI638	Statistics and Data Science	2	0	0	2	2
18	Professional Elective	19B1WCI678	Statistics and Data Science Lab	0	0	2	1	2

ELECTIVE-III

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI638	Pattern Recognition	2	0	0	2	2
2	Professional Elective	18B1WCI635	Data Mining & Data Warehousing	2	0	0	2	2
3	Professional Elective	18B1WCI632	Parallel and Distributed Algorithms	2	0	0	2	2
4	Professional Elective	18B1WCI636	Digital Image processing	2	0	0	2	2
5	Professional Elective	18B1WCI678	Pattern Recognition Lab	0	0	2	1	2
6	Professional Elective	18B1WCI675	Data Mining & Data Warehousing Lab	0	0	2	1	2
7	Professional Elective	18B1WCI672	Parallel and Distributed Algorithms Lab	0	0	2	1	2
8	Professional Elective	18B1WCI676	Digital Image processing Lab	0	0	2	1	2
9	**Professional Elective**	19B1WCI632	**Information Security**	**2**	**0**	**0**	2	2
10	**Professional Elective**	19B1WCI672	**Information Security Lab**	**0**	**0**	**2**	1	2
11	**Professional Elective**	20B1WCI732	**From Graph to Knowledge Graph**	**2**	**0**	**0**	2	2
12	**Professional Elective**	20B1WCI772	**From Graph to Knowledge Graph Lab**	**0**	**0**	**2**	1	2
13	**Professional Elective**	19B1WCI635	**Architecting Distributed Cloud Applications**	**2**	**0**	**0**	2	2
14	**Professional Elective**	19B1WCI675	**Architecting Distributed Cloud Applications Lab**	**0**	**0**	**2**	1	2
15	**Professional Elective**	19B1WCI637	**Statistics and Exploratory Data Analytics**	**2**	**0**	**0**	2	2
16	**Professional Elective**	19B1WCI677	**Statistics and Exploratory Data Analytics Lab**	**0**	**0**	**2**	1	2

ELECTIVE-IV

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI734	Cryptography& network security	2	0	0	2	2
2	Professional Elective	18B1WCI743	Advanced Algorithms	2	0	0	2	2
3	Professional Elective	18B1WCI741	R-Programming	2	0	0	2	2
4	Professional Elective	18B1WCI742	Artificial Intelligence	2	0	0	2	2
5	Professional Elective	18B1WCI774	Cryptography& network security Lab	0	0	2	1	2
6	Professional Elective	18B1WCI773	Advanced Algorithms Lab	0	0	2	1	2
7	Professional Elective	18B1WCI771	R-Programming Lab	0	0	2	1	2
8	Professional Elective	18B1WCI772	Artificial Intelligence Lab	0	0	2	1	2
9	**Professional Elective**	19B1WCI731	**Computational Data Analysis**	**2**	**0**	**0**	2	2
10	**Professional Elective**	19B1WCI771	**Computational Data Analysis lab**	**0**	**0**	**2**	1	2
11	**Professional Elective**	19B1WCI732	**Game Development and Design**	**2**	**0**	**0**	2	2
12	**Professional Elective**	19B1WCI772	**Game Development Lab**	**0**	**0**	**2**	1	2

ELECTIVE-V

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI736	Storage Networks	3	0	0	3	3
2	Professional Elective	18B1WCI738	Internet of Things	3	0	0	3	3
3	Professional Elective	18B1WCI735	Mobile Computing	3	0	0	3	3
4	Professional Elective	18B1WCI737	Cloud Computing	3	0	0	3	3
5	Professional Elective	18B1WCI740	Computational Techniques and Algorithms in Engineering	3	0	0	3	3
6	Professional Elective	19B1WCI736	Information Auditing & Risk Management	3	0	0	3	3
7	Professional Elective	19B1WCI737	Optimization Methods in Business Analytics	3	0	0	3	3
8	Professional Elective	18B1WCI840	Computer Vision	3	0	0	3	3
9	Professional Elective	19B1WCI738	Introduction to Deep Learning	3	0	0	3	3

ELECTIVE-VI

S. No.	Category Code	Subject Code	Name of the Subjects	Course Hours			Credits	Total Hours
				L	T	P		
1	Professional Elective	18B1WCI843	Data Analytics	3	0	0	3	3
2	Professional Elective	18B1WCI844	BIG DATA	3	0	0	3	3
3	Professional Elective	18B1WCI845	Network Management	3	0	0	3	3
4	Professional Elective	18B1WCI846	Graph Theory	3	0	0	3	3
5	Professional Elective	19B1WCI831	Ethics and Information Technology	3	0	0	3	3
6	Professional Elective	18B1WCI832	Social and Information Network Analysis	3	0	0	3	3
7	Professional Elective	19B1WCI832	Probabilistic Graphical Models	3	0	0	3	3
8	Professional Elective	19B1WCI833	Information Modeling	3	0	0	3	3
9	Professional Elective	19B1WCI834	Information Visualization	3	0	0	3	3
10	Professional Elective	19B1WCI835	Cloud Computing Security	3	0	0	3	3
11	Professional Elective	19B1WCI836	Knowledge-Based AI: Cognitive Systems	3	0	0	3	3
12	Professional Elective	19B1WCI837	Reinforcement Learning	3	0	0	3	3

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, SOLAN								
COURSE CURRICULUM OF DEPARTMENT CSE&IT - 2021 Batch (160 CREDITS)								
LIST OF OPEN ELECTIVES								
S. No.	**Category Code**	**Semester**	**Name of the Subjects**	**Course Hours**			**Credits**	**Total Hours**
				L	**T**	**P**		
OPEN ELECTIVE II (7th Semester)								
1	19B1WCI733	7	Introduction to C++ Programming	2	0	0	2	2
2	19B1WCI734	7	Object-Oriented Technologies using Java	2	0	0	2	2
3	19B1WCI735	7	Software Testing Methodologies	2	0	0	2	2
4	19B1WCI773	7	Introduction to C++ Programming Lab	0	0	2	1	2
5	19B1WCI774	7	Object-Oriented Technologies using Java Lab	0	0	2	1	2
6	19B1WCI775	7	Software Testing Methodologies Lab	0	0	2	1	2
OPEN ELECTIVE III (7th Semester)								
1	13B1WCI731	7	ARM based Embedded System Design	3	0	0	3	3
2	19B1WCI739	7	Software Defined Network	3	0	0	3	3
3	19B1WCI740	7	Introduction to Statistical learning	3	0	0	3	3
OPEN ELECTIVE IV (8th Semester)								
1	19B1WCI838	8	Principles of Distributed Database Systems	3	0	0	3	3
2	19B1WCI839	8	Foundations of Blockchain	3	0	0	3	3
3	19B1WCI840	8	Computational Biology	3	0	0	3	3
OPEN ELECTIVE V (8th Semester)								
1	19B1WCI841	8	Wireless Sensor Networks: Protocols and Applications	3	0	0	3	3
2	19B1WCI842	8	Service Oriented Architecture	3	0	0	3	3
3	19B1WCI843	8	Multimedia Systems and Applications	3	0	0	3	3

B.TECH. COMPUTER SCIENCE AND ENGINEERING

SYLLABUS

Approved in Academic Council held on 28 July 2021

Programming for Problem Solving-II

COURSE CODE: 19B11CI111

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 2-0-0

Pre-requisite: None

Course Objectives:

1. To formulate simple algorithms for arithmetic and logical problems.
2. To translate the algorithms to programs (in C language).
3. To test and execute the programs and correct syntax and logical errors.
4. To implement conditional branching, iteration and recursion.
5. To decompose a problem into functions and synthesize a complete program using divide andconquer approach.
6. To use arrays, pointers and structures to formulate algorithms and programs.
7. To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
8. To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration

Course outcomes:

S.NO	Course outcomes	Level of Attainment
CO-1	To formulate simple algorithms for arithmetic and logical problems.	Familiarity
CO-2	To translate the algorithms to programs (in C language).	Familiarity
CO-3	To test and execute the programs and correct syntax and logical errors.	Usage
CO-4	To implement conditional branching, iteration and recursion.	Usage
CO-5	To decompose a problem into functions and synthesize a complete program using divide and conquer approach.	Usage
CO-6	To use arrays, pointers and structures to formulate algorithms and programs.	Usage
CO-7	To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.	Assessment
CO-8	To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Programming (4 lectures)** Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.) - (1 lecture). **Idea of Algorithm:** steps to solve logical and numerical problems. Representation of Algorithm: Flowchart/Pseudocode with examples. (1 lecture)	4

	From algorithms to programs; source code, variables (with data types) variables and memory locations, Syntax and Logical Errors in compilation, object and executable code- (2 lectures)	
2	**Arithmetic expressions and precedence**	**2**
3	**Loops:** Conditional Branching and Loops (6 lectures) Writing and evaluation of conditionals and consequent branching (3 lectures) Iteration and loops (3 lectures)	**6**
4	**Arrays:** Arrays (1-D, 2-D), Character arrays and Strings	**6**
5	**Basic Algorithms:** Searching, Basic Sorting Algorithms (Bubble, Insertion and Selection), Finding roots of equations, notion of order of complexity through example programs (no formal definition required.	**6**
6	**Function:** Functions (including using built in libraries), Parameter passing in functions, call by value, Passing arrays to functions: idea of call by reference **Recursion:** Recursion, as a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series, Ackerman function etc. Quick sort or Merge sort.	**5** **4**
7	**Structure:** Structures, Defining structures and Array of Structures	**4**
8	**Pointers:** Idea of pointers, Defining pointers, Use of Pointers in self- referential structures, notion of linked list (no implementation) **File handling**	**3** **2**
Total lectures		**42**

Suggested Text Book(s):

1. Byron Gottfried, Schaum's Outline of Prokli[gramming with C, McGraw-Hill
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill

Suggested Reference Book(s):

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc18-cs10

2. Link to topics related to course:
   a. **https://www.learn-c.org/**
   b. **https://www.programiz.com/c-programming**
   c. **https://www.codechef.com/ide**

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Programming for Problem Solving)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	2	3	2	2	3	2	3	2	2	3	3	**2.5**
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	**2.6**
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	**2.5**
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	**2.4**
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	**2.6**
CO-7	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
CO-8	3	2	3	2	2	3	2	3	2	2	3	3	**2.5**
Average	**2.6**	**2.1**	**2.6**	**2.1**	**2.4**	**2.6**	**2.5**	**2.6**	**2.1**	**2.5**	**2.9**	**2.8**	

Programming for Problem Solving Lab-II

COURSE CODE: 19B17CI171

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: No prior programming experience is expected however, mathematical maturity level of science or engineering undergraduate is assumed.

Course Objectives:

1. Develop problem solving ability using programming.
2. To impart adequate knowledge on the need of programming languages and problem solving techniques.
3. To develop a methodological way of problem solving
4. Analyze and construct effective algorithms
5. Employ good programming practices such as incremental development, data integrity checking and adherence to style guidelines
6. Learn a programming approach to solve problems

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	Understand the Typical C Program Development Environment, compiling, debugging, Linking and executing.	Familiarity
CO-2	Introduction to C Programming using Control Statements and Repetition Statement	Usage
CO-3	Apply and practice logical formulations to solve some simple problems leading to specific applications.	Assessment and Usage
CO-4	Design effectively the required programming components that efficiently solve computing problems in real world.	Assessment & Usage

List of Experiments:

S.No	**Description**	**Hours**
1	Getting acquainted with the C program Structure and basic I/O. Getting acquainted with the various data types and arithmetic operator used in C.	2
2	Write a program to obtain the reversed number and to determine whether the original and reversed numbers are equal or not.	2
	Write a program to check whether a triangle is valid or not, when the three angles of triangle are entered through the keyboard. A triangle is valid if the sum of all three angles is equal to 180 degrees. Check a given I/P is character, number or special symbol.	
3	WAP to check a given number is Armstrong or not. Calculate factorial of a number Given number is prime or not.	2
4	Write a program to add first seven terms of the following series using any loop: 1/1! + 2/2! + 3/3! +….. Any five pattern program.	2
5	WAP to swap two numbers with function using 3rd variable or without using (call by value & reference).	2

	Write a function to find out the roots of quadratic equation.	
6	Factorial using recursion Fibonacci series using recursion.	2
7	WAP to sort N elements of an array using bubble sort. WAP for Binary search & linear search.	2
8	Find Max, Min, 2nd Max, Standard Deviation. Reverse elements of an array.	2
9	Matrix addition, Multiplication and Transpose.	2
10	WAP to handle pointer variables and access the elements of an array using pointers. WAP to insert a string and perform operations: string length, copy, concatenation, compare, lower to upper, etc.	2
11	Write a program to find whether the string is palindrome or not using pointers Write a program to delete all vowels from sentence, assume that sentence is not more than 80 character long using pointers.	2
12	Enter the detail of 5 students using structure and print the details of all students including pointers and also sort the detail of students using DOB.	2
13	Dynamic allocation function and random function with string and integer array.	2
14	Perform operation on files: open, read, write, close etc.	2
	Total Lab hours	**28**

Suggested/Resources:

1. Yale N. Patt and Sanjay J. Patel, Introduction to Computing Systems, from bits & gates to C & beyond, 2nd Edition, 2004.
2. Deitel and Deitel, C How to Program, 7th Edition, 2013.
3. Venugopal Prasad, Mastering C, Tata McGraw Hill.
4. Complete Reference with C, Tata McGraw Hill.
5. Drmey, How to solve it by Computer, PHI.
6. Kerninghan and Ritchie, The C Programming Language.
7. http://www.acm.uiuc.edu/webmonkeys/book/c_guide/
8. http://msdn.microsoft.com/en-us/library/25db87se.aspx

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	Average
CO1	3	3	1	1	2	2	1	1	1	2	1	2	**1.7**
CO2	3	3	2	1	3	1	1	1	1	2	1	2	**1.8**
CO3	3	3	2	2	2	3	2	1	1	2	2	2	**2.1**

CO4	3	3	3	3	3	2	1	1	1	2	1	3	**2.2**
Average	**3**	**3**	**2**	**1.8**	**2.5**	**2**	**1.3**	**1**	**1**	**2**	**1.3**	**2.3**	

Data Structure and Algorithms

COURSE CODE: 18B11CI211

COURSE CREDIT: 4

CORE/ELECTIVE: CORE

: 3-1-0

Pre-requisites: C/C++

Course Objectives:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures
5. Introduce students to data abstraction and fundamental data structures.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	To gain knowledge on the notions of data structure, Abstract Data Type.	Familiarity
CO-2	For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness.	Assessment
CO-3	For a given Search problem (Linear Search and Binary Search) student will able to implement it.	Assessment
CO-4	For a given problem of Stacks, Queues and linked list student will able to implement it and analyze the same to determine the time and computation complexity.	Assessment
CO-5	Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity.	Assessment
CO-6	Student will able to write an algorithm Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of Space and Time complexity.	Usage

Course Contents:

Unit	Contents	Lectures required

1	**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis.	**7**
2	**Stacks:** ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis.	**5**
3	**Queues:** ADT queue, Types of Queue: Simple Queue, Circular Queue, Priority Queue; Operations on each types of Queues: Algorithms and their analysis.	**5**
4	**Linked Lists:** Singly linked lists: Representation in memory, Algorithms of several operations: Traversing, Searching, Insertion into, Deletion from linked list; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists: all operations their algorithms and the complexity analysis.	**8**
5	**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree; Tree operations on each of the trees and their algorithms with complexity analysis. Applications of Binary Trees. B Tree, B+ Tree: definitions, algorithms and analysis.	**6**
6	**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods, Hashing.	**6**
7	**Graph:** Basic Terminologies and Representations, Graph search and traversal algorithms and complexity analysis.	**5**
Total lectures		**42**

Suggested Text Book(s):

1. "Fundamentals of Data Structures", Illustrated Edition by Ellis Horowitz, Sartaj Sahni, Computer Science Press

Suggested Reference Book(s):

1. "Algorithms, Data Structures, and Problem Solving with C++", Illustrated Edition by Mark Allen Weiss, Addison-Wesley Publishing Company
2. "How to Solve it by Computer", 2nd Impression by R. G. Dromey, Pearson Education.
3. "Data structures and Algorithms Made Easy" 5th edition by Narasimha Karumanchi, Career monk publications
4. "Data Structure and Algorithms in C" 2nd edition by Mark Allen Weiss (2002), Pearson Education

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106102064/
2. Link to topics related to course:
   - **a.** https://onlinecourses.nptel.ac.in/noc18_cs25/preview
   - **b.** https://nptel.ac.in/courses/106103069/

c. http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course Outcomes (Data Structure and Algorithms)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Data Structure and Algorithms Lab

COURSE CODE: 18B17CI271

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisites: None

Course Objectives:

1. Develop problem solving ability using Programming
2. Develop ability to design and analyze algorithms
3. Introduce students to data abstraction and fundamental data structures
4. Develop ability to design and evaluate Abstract Data Types and data structures
5. Apply data structure concepts to various examples and real life applications

Course outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	To gain knowledge on the notions of data structure, Abstract Data Type	Familiarity
CO-2	To have hands on skills to evaluate different kinds of linked lists and their applications in day to day problem solving.	Usage
CO-3	To have hands on skills to evaluate different kinds stacks and their applications and implementations in day to day problem solving	Assessment
CO-4	To have hands on skills to evaluate different kinds queues and their applications and implementations in simulations.	Assessment
CO-5	To acquire knowledge of various sorting algorithms	Usage
CO-6	To learn Searching: Balanced tree, red-black tree, lower bounds for searching	Usage
CO-7	To learn to code for operations on Tree or BST (Creation; Traversing like pre-order,post-order and in-order; Searching element; finding height etc.)	Usage
CO-8	Introduction to Heaps	Usage
CO-9	To learn to code for operations on Graphs (Creation; entering info, printing output and deleting; traversal of BFS and DFS algorithm)	Assessment

List of Experiments:

S.No	**Description**	**Hours**
1	**Getting acquainted with** a) Arrays and Strings, Structures, b) Recursion, Pointers c) Dynamic memory allocation	 2 4 4
2	**Operations on: (Creation, insertion, deletion, sorting, traversing, reversing etc)** a) Linear Linked List, b) Doubly and c) Circular Linked List	 4 4 2

3	**Operations on Stacks:**
	a) Creation; pushing; popping;
	b) testing underflow, overflow;
	c) prefix and postfix
		
		4
		2
		2

4	**Operations on Queues:**
	a) Creation;
	b) enqueue; dequeue;
	c) testing underflow, overflow
		
		4
		2
		2

5	**Operations on Tree or BST:** Creation; a) Traversing like preorder, post-order and in-order; b) Searching element; finding height etc.	 4 2
6	**Implementation of sorting algorithms 1:** Insertion Sort and Selection Sort Algorithm with arrays using dynamic memory allocation.	2
7	**Implementation of sorting algorithms 2:** Bubble Sort and Merge Sort Algorithm with arrays using dynamic memory allocation.	2
8	**Implementation of sorting algorithms 3:** Implementation of Radix Sort and Quick Sort Algorithm with arrays using dynamic memory allocation.	2
9	**Operation on Heaps:** a) Heaps, b) Heap Sort	 2 2
10	**Implementation of Searching algorithms:** Linear Search Algorithm and Binary Search Algorithm using dynamic memory allocation.	2
11	**Operations on Graphs :** (Creation; entering info; printing Output and deleting; traversal of BFS and DFS algorithm etc.)	2
Total Lab hours		**56**

Minor Project(s) – (*Only for 2 credit lab)*

- Design GUI based program to solve any binary equation.
- Design GUI based program to find the roots of quadratic equation.

- Design a program that picks the characters at equal interval from the given text/paragraph and generate a new paragraph in which each set of word can’t have more than 4 characters. Last word of the paragraph can have <=4 characters.
- Program to input following data into disk file. Code, name, department and salary of employee in a firm. After creating file read the file and find following-
  Methodology
  algorithms
  Code execution
  Future scope
  Count number of employees as per department
  Search record of employee
  Display record of employee
  Display list of employee in alphabetical order as per department

Read record from file

Suggested Books/Resources:

1. Langsam, Augestein, Tenenbaum : Data Structures using C and C++, 2nd Edn, 2000, Horowitz and Sahani : Fundamental of Data Structures in C, 2nd Edn, 2008
2. Weiss : Data Structures and Algorithm Analysis in C/C++, 3rd Edn, 2006
3. Sahani : Data Structures, Algorithms and applications in C++, 1997.
4. Corman et al : Introduction to Algorithms, 3rd Edn., 2009
5. http://www.nptel.iitm.ac.in/video.php?subjectId=106102064, last accessed Mar 13, 2014.
6. http://www.cs.auckland.ac.nz/~jmor159/PLDS210/ds_ToC.html, last accessed Mar 13, 2014.
7. http://courses.cs.vt.edu/csonline/DataStructures/Lessons/index.html, last accessed Mar 13, 2014.
8. Link to topics related to course:
   a. http://cse.iitkgp.ac.in/~pallab/pds16/pds16.htm
   b. https://onlinecourses.nptel.ac.in/programming101/preview

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	3	2	2	3	3	3	2	3	2	2	**2.5**
CO-2	3	3	3	2	3	3	3	3	2	3	2	3	**2.8**
CO-3	3	3	3	2	2	3	3	3	3	3	2	2	**2.7**
CO-4	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-5	3	3	3	2	2	2	3	3	3	3	2	2	**2.6**
CO-6	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-7	3	3	3	3	3	3	2	2	3	3	3	3	**2.8**
CO-8	3	3	3	2	3	3	3	3	3	3	2	3	**2.8**
CO-9	3	3	2	3	3	3	3	3	3	2	3	3	**2.8**
Average	**3**	**2.9**	**2.9**	**2.4**	**2.7**	**2.9**	**2.9**	**2.7**	**2.6**	**2.9**	**2.4**	**2.7**	

Python Programming Essentials

COURSE CODE: 18B11CI314

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. Provide an understanding of the role computation can play in solving problems.
2. Help students, including those who do not plan to major in Computer Science and Electrical Engineering (like BI and BT), feel confident of their ability to write small programs that allow them to accomplish useful goals.
3. Position students so that they can compete for research projects and excel in subjects with programming components.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Familiarity about concepts of Python Programming. Broaden the knowledge about Variables, expressions and Functions in Python.	Familiarity
CO-2	Broaden the knowledge about Branching and Iteration. To have hands on skills on String Manipulation, Guess and Check, Approximations, Bisection	Assessment
CO-3	To learn about Decomposition Abstractions, Tuples, Lists, Dictionaries and Illustrative programs:	Assessment
CO-4	Files, Modules, Packages and Testing, Debugging, Exceptions, Assertions:	Usage
CO-5	Understanding and analyzing Object Oriented Programming:	Familiarity
CO-6	To have hands on skills on Illustrative programs(examples Sorting and Searching, Regular expressions) and GUI	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Informal introduction to Python programming language:** What is a program?, What is debugging?, Formal and natural languages, Downloading and installing Python., The first program, Debugging **Variables, expressions and Functions in Python** Values and types, Variables, Variable names and keywords, Operators and operands, Expressions and statements, Interactive mode and script mode, Order of operations, String operations, Function calls, Type conversion functions, Math functions, Composition, Adding new functions, Definitions and uses, Flow of execution **Parameters and arguments**	5
2	**Branching and Iteration:** Loops, Multiple assignment, Updating variables, The while statement, Break **String Manipulation, Guess and Check, Approximations, Bisection:** String manipulation, Guess and check algorithms(e.g: find Square Root etc), Approximate solutions(e.g	8

	Successive approximation), Bisection method.	
3	**Decomposition, Abstractions:** divide and conquer (modules), Abstraction **Tuples, Lists, Dictionaries, Illustrative programs:** - Lists: list operations, list slices, list methods, list loop, mutability, aliasing, cloning lists, list parameters; - Tuples: tuple assignment, tuple as return value; - Dictionaries: operations and methods; advanced list processing – list comprehension; Illustrative programs: selection sort, insertion sort, mergesort, histogram	**7**
4	**Files, Modules, Packages**: Files and exception: text files, reading and writing files, format operator, Packages, Illustrative programs: word count, copy file. **Testing, Debugging, Exceptions, Assertions:** Unit testing framework(unit test), debugger for Python program(pdb), Handling an exception	**7**
5	**Object Oriented Programming:** -Classes, objects, attributes and methods; defining classes; design with classes, data modeling; persistent storage of objects -OOP, continued: inheritance, polymorphism, operator overloading, abstract classes.	**7**
6	**Illustrative programs(examples):** • Sorting and Searching • Regular expressions o Match function o Search function o Matching vs. Searching o Modifiers o Patterns **GUI :** Introduction, Tkinter programming, Tkinter widgets	**8**
Total lectures		**42**

Suggested Text Book(s):

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning, ISBN: 978-1111822705

Suggested Reference Book(s):

1. Kenneth A. Lambert, The Fundamentals of Python: First Programs, 2011, Cengage Learning, ISBN: 978-1111822705.
2. Chun, Wesley. Core python programming. Vol. 1. Prentice Hall Professional, 2001.
3. Zelle, John M. Python programming: an introduction to computer science. Franklin, Beedle & Associates, Inc., 2004.
4. Gold, Steve. "Python: Python Programming Learn Python Programming In A Day-A Comprehensive Introduction To The Basics Of Python & Computer Programming." (2016).

Other useful resource(s):

1. https://onlinecourses.nptel.ac.in/noc18_cs35/preview
2. https://nptel.ac.in/courses/106106145/

3. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/index.htm
4. https://docs.python.org/3/tutorial/index.html

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	*T-1	15	1 Hour.	Syllabus covered up to T-1
2	#T-2	25	1.5 Hours	Syllabus covered up to T-2
3.	$T-3	35	2 Hours	Entire Syllabus
4.	~Internal Assessment	25	Entire Semester	Assignment Quizzes Attendance

*In Test-I Paper the 20% of 15 Marks will be allocated to Introduction to Computers portion and 80% of 15 Marks will be allocated to Introduction to Programming portion.

#In Test-II Paper the 20% of 25 Marks will be allocated to syllabus of Test-I and 80% of 25 Marks will be allocated to further covered portion.

$In Test-III Paper the 40% of 30 Marks will be allocated to syllabus of Test-I+ Test-II and 60% of 30 Marks will be allocated to further covered portion.

~Internal Assessment will purely be focused on the assignments and quizzes based on Python Programming.

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Python Programming Essentials)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	2	3	1	2	3	3	2	1	1	1	3	**2.1**
CO-2	3	3	3	1	2	3	3	3	2	1	1	3	**2.3**
CO-3	3	3	3	1	2	3	2	3	2	1	1	3	**2.3**
CO-4	3	3	3	3	3	3	3	2	2	1	1	3	**2.5**
CO-5	3	3	3	2	2	3	3	3	2	1	1	3	**2.4**
CO-6	3	3	3	2	3	2	3	2	2	1	1	3	**2.3**
Average	**3**	**2.8**	**3**	**1.7**	**2.3**	**2.8**	**2.8**	**2.5**	**1.8**	**1**	**1**	**3**	

Object-Oriented Systems and Programming

COURSE CODE: 18B11CI311

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

1. To use object modelling technique to analyze problem requirements, design a solution to the problem andthen implement the solution in Object-Oriented Programming Language(s) or database.
2. To strengthen their problem solving ability by applying the characteristics of an object-orientedapproach.
3. To strengthen ability to design and represent solutions to problems using UMLnotations
4. To introduce object oriented concepts in C++ and Java.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	To learn the concepts of Objects, Classes, Methods, Constructors and Destructors	Usage
CO-2	To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions.	Usage
CO-3	To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-levelInheritance, Hierarchical Inheritance and Hybrid Inheritance.	Usage
CO-4	To learn the concept of Abstract classes and interfaces	Usage
CO-5	To learn the concepts of Operator overloading and conversion function	Usage
CO-6	To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file.	Usage
CO-7	To learn the Exception Handling: try --catch and finally block, making user-defined exceptions.	Usage
CO-8	To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	Structured versus Object-Oriented programming, Object-Oriented paradigm. Defining Objects, Classes, Data members, Member functions, Constructors and Destructors in C++. Inline Functions, Friend Functions, Constant member functions, and Static members (static data and static member functions). Function overloading, Operator overloading and Conversion functions. Using the concepts of File handling	**12**
2	Polymorphism and Inheritance in C++. Abstract classes, virtual function, pure virtual functions, and virtual base classes in C++.	**6**
3	Function templates and Class templates	**3**
4	Introduction to Java and its features. Defining Classes in Java, Wrapper classes, Packages and Exception handling in Java	**7**
5	Inheritance and Interfaces in Java. Abstract class, abstract methods,	**5**

	final class and final method in Java	
6	Basic principles of Software engineering. System analysis, design, testing and debugging. Unified Modeling Language (Class Diagram, Use Case Diagram, State Diagram, Sequenced Diagram, Communication Diagram, Activity Diagram)	9
Total lectures		**42**

Suggested Text Book(s):

1. Lafore R., Object oriented programming in C++, Waite Group
2. Java 2: The Complete Reference, Fifth Edition -- by Herbert Schildt
3. Satzinger, Jackson, Burd, Object-Oriented Analysis & Design

Suggested Reference Book(s):

1. Stroustrap B., The C++ Programming Language, Addison Wesley
2. Bruce Eckel, Thinking in C++
3. Bruce Eckel, Thinking in Java

Other useful resource(s):

1. Link to NPTEL course contents:

   a. https://onlinecourses.nptel.ac.in/noc16_cs17/preview
   b. http://www.nptelvideos.com/java/java_video_lectures_tutorials.php
   c. https://onlinecourses.nptel.ac.in/noc17_cs25/announcements

2. Link to topics related to course:

   a. https://www.tutorialspoint.com/cplusplus/
   b. http://www.cplusplus.com/doc/tutorial

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Internal Assessment	25	Entire Semester	Assignment Quizzes Attendance

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Object-Oriented Systems and Programming)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	3	2	2	3	3	2	3	3	3	2.8

CO-2	3	3	3	2	1	1	2	3	2	3	3	3	**2.4**
CO-3	3	3	2	2	2	1	2	2	2	3	3	3	**2.3**
CO-4	3	3	3	2	2	1	2	3	2	3	3	3	**2.5**
CO-5	3	3	3	3	2	1	2	3	2	1	3	3	**2.4**
CO-6	3	3	3	3	2	3	2	3	2	2	3	3	**2.7**
CO-7	3	3	2	2	2	3	3	2	2	3	3	3	**2.6**
CO-8	3	3	3	3	3	2	3	3	2	1	3	1	**2.5**
Average	**3**	**3**	**2.8**	**2.5**	**2**	**1.8**	**2.4**	**2.8**	**2**	**2.4**	**3**	**2.8**	

Database Management Systems

COURSE CODE: 18B11CI313

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

: 3-0-0

Pre-requisite: Introduction to Computer Programming, Discrete Mathematics, Data Structures

Course Objectives:

1. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models
2. To understand and use data manipulation language to query, update, and manage a database.
3. To develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, and intelligent database, Client/Server (Database Server), Data Warehousing.
4. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Explain the characteristics, architecture of database approach, its components, different data models and the examples of their usage.	Familiarity
CO-2	For a given query write relational algebra expressions for that query and optimize the developed expressions.	Usage
CO-3	For a given specification of the requirement, design the databases using E-R method and normalization.	Usage
CO-4	Determine the functional dependency between two or more attributes, compute the closure of a set of attributes, evaluate a proposed decomposition	Assessment
CO-5	Give examples of the application of primary, secondary, and clustering indexes, explain the theory and application of internal and external hashing techniques.	Assessment
CO-6	Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.	Assessment
CO-7	Familiarize with the security in databases and gaining familiarity with other popular databases used in the industry	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	**Database system architecture:** Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). **Data models:** Entity-relationship model, network model, relational and object oriented data models, integrity constraints, data manipulation operations.	**5**
2	**Relational query languages:** Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS -	**15**

	MYSQL, ORACLE, DB2, SQL server. **Relational database design:** Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design. **Query processing and optimization:** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.	
3	**Storage strategies:** Indices, B-trees, hashing.	**6**
4	**Transaction processing:** Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.	**9**
5	**Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.	**5**
6	**Advanced topics:** Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.	**3**
Total lectures		**42**

Suggested Text Book(s):

1. "Fundamentals of Database Systems" Elmasri, Navathe, Pearson Education.
2. "Database system concepts" Henry F Korth, Abraham Silberschatz, S. Sudurshan, McGraw-Hill Ellis

Suggested Reference Book(s):

1. "Database Systems: A Practical Approach to design, Implementation and Management". Thomas Connolly, Carolyn Begg; Third Edition, Pearson Education.
2. Bipin C Desai, ?An Introduction to Database Systems?, Galgotia. Publications Pvt Limited, 2001
3. "An Introduction to Database Systems", C.J.Date, Pearson Education.
4. "A first course in Database Systems", Jeffrey D. Ullman, Jennifer Windon, Pearson, Education.
5. "Data Management: databases and organization", Richard T. Watson, Wiley.
6. " Data Modeling Essentials", Graeme C. Simxion, Dreamtech.

.

Other useful resource(s):

1. Link to NPTEL course contents: https://www.youtube.com/watch?v=EUzsy3W4I0g&list=PL9426FE14B809CC41
2. Link to topics related to course:
   - **a.** https://www.tutorialspoint.com/dbms/database_normalization.htm
   - **b.** https://www.igi-global.com/journal/journal-database-management/1072
   - **c.** https://www.tutorialspoint.com/dbms/dbms_hashing.htm

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Internal Assessment	25	Entire Semester	Assignment Quizzes Attendance

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Database Management Systems)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	2	1	2	2	2	1	2	3	3	2	**2.2**
CO-2	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
CO-3	3	3	2	1	1	3	3	3	3	3	1	1	**2.3**
CO-4	3	2	3	1	2	2	2	2	1	2	1	2	**1.9**
CO-5	3	2	2	1	2	3	3	2	1	3	2	1	**2.1**
CO-6	3	2	3	1	1	3	2	1	1	3	2	1	**1.9**
CO-7	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
Average	**3**	**2.6**	**2.4**	**1.1**	**1.6**	**2.6**	**2.6**	**1.7**	**1.9**	**2.6**	**2**	**1.4**	

Python Programming Lab

COURSE CODE: 18B17CI374

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisite: None

Course Objectives:

1. Learn syntax, semantics and create Functions in Python.
2. Understand the usage of Lists, Dictionaries, and arrays in Python.
3. Learn the Implementation of object oriented programming concepts in Python
4. Learn different data structure in Python.
5. Understand file handling in Python
6. Implement GUI applications and browser.

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO1	Familiarity about concepts of Python Programming. Broaden the knowledge about Variables, expressions and Functions in Python.	Familiarity
CO2	Broaden the knowledge about Branching and Iteration. To have hands on skills on String Manipulation, Guess and Check, Approximations, Bisection	Assessment
CO3	To learn about Decomposition Abstractions, Tuples, Lists, Dictionaries and Illustrative programs:	Assessment
CO4	Files, Modules, Packages and Testing, Debugging, Exceptions, Assertions:	Usage
CO5	Understanding and analyzing Object Oriented Programming:	Familiarity
CO6	To have hands on skills on Illustrative programs(examples Sorting and Searching, Regular expressions) and GUI	Usage

List of Experiments

S.No.	**Description**	**Hours**
1	1. Write a Python program to get the Python version you are using. 2. Write a Python program which accepts the radius of a circle from the user and compute the area.	**2**
2	1. Write a Python program to display the current date and time. 2. Write a Python program which accepts the radius of a circle from the user and compute the area.	2
3	1. Write a Python program which accepts the user's first and last name and print them in reverse order with a space between them. 2. Write a Python program to display the first and last colors from the following list. color_list = ["Red","Green","White" ,"Black"].	2
4	1. Write a Python program to print the documents (syntax, description etc.) of Python built-in function(s). Samplefunction :abs() ExpectedResult : abs(number)->number Return the absolute value of the argument.	2
5	Write a Python program to get the difference between a given number and 17, if the	2

	number is greater than 17 return double the absolute difference. Write a Python program to test whether a number is within 100 of 1000 or 2000. Write a Python program to check whether a specified value is contained in a group of values. Test Data : 3 -> [1, 5, 8, 3] : True -1 -> [1, 5, 8, 3] : False	
6	Write a Python program to print all even numbers from a given numbers list in the same order and stop the printing if any numbers that come after 237 in the sequence. Sample numbers list : numbers = [386, 462, 47, 418, 907, 344, 236, 375, 823, 566, 597, 978, 328, 615, 953, 345, 399, 162, 758, 219, 918, 237, 412, 566, 826, 248, 866, 950, 626, 949, 687, 217, 815, 67, 104, 58, 512, 24, 892, 894, 767, 553, 81, 379, 843, 831, 445, 742, 717, 958,743, 527] Now do it using file as input.	**2**
7	Write a Python program that accepts a single integer value entered by the user. If the value entered is less than one, the program prints nothing. If the user enters a positive integer, n, the program prints an n×n box drawn with * characters. If the users enters 1, for example, the program prints * If the user enters a 2, it prints ** ** An entry of three yields.	**2**
8	Write a Python program to sum of two given integers. However, if the sum is between 15 to 20 it will return 20. Write a Python program to compute the future value of a specified principal amount, rate of interest, and a number of years. Test Data : amt = 10000, int = 3.5, years = 7 Expected Output : 12722.79	**2**
9	Write a Python program to create an array of 5 integers and display the array items. Access individual element through indexes. Write a Python program to convert an array to an ordinary list with the same items.	**2**
10	Write a Python program to display all the member name of an enum class ordered by their values. Expected Output: Country Name ordered by Country Code: Afghanistan Algeria Angola Albania Andorra Antarctica	**2**
11	Write a Python program to get all values from an enum class. Expected output: [93, 355, 213, 376, 244, 672].	**2**
12	Write a Python program to get an array buffer information Expected Output: Array buffer start address in memory and number of elements. (25855056, 2)	**2**
13,14	Write a Python program to push three items into a heap and return the smallest item from the heap. Also Pop and return the smallest item from the heap ExpectedOutput: Items in the heap: ('V', 1) ('V', 3) ('V', 2) ------------------------ The smallest item in the heap: ('V', 1) ------------------------ Pop the smallest item in the heap: ('V', 2) ('V', 3)	**4**
15,16	Write a function named print_big_enough that accepts two parameters, a list of numbers and a number. The function should print, in order, all the elements in the list that are at least as large as the second parameter.	**4**

17,18	Write a function called draw_rectangle that takes a Canvas and a Rectangle as arguments and draws a representation of the Rectangle on the Canvas. 2. Add an attribute named color to your Rectangle objects and modify draw_rectangle so that it uses the color attribute as the fill color. 3. Write a function called draw_point that takes a Canvas and a Point as arguments and draws a representation of the Point on the Canvas. 4. Define a new class called Circle with appropriate attributes and instantiate a few Circle objects. Write a function called draw_circle that draws circles on the canvas. 5. Write a program that draws the national flag of the India. Hint: you can draw a polygon like this: points = [[-150,-100], [150, 100], [150, -100]] canvas.polygon(points, fill='saffron,white,green')	**4**
19,20	The datetime module provides date and time objects that are similar to the Date and Time objects in this chapter, but they provide a rich set of methods and operators. Read the documentation at http: // docs. python. org/ 2/ library/ datetime. html . 1. Use the datetime module to write a	**4**
	program that gets the current date and prints the day of the week. 2. Write a program that takes a birthday as input and prints the user‟s age and the number of days, hours, minutes and seconds until their next birthday. 3. For two people born on different days, there is a day when one is twice as old as the other. That‟s their Double Day. Write a program that takes two birthdays and computes their Double Day. 4. For a little more challenge, write the more general version that computes the day when one person is n times older than the other.	
21,22	This exercise is a cautionary tale about one of the most common, and difficult to find, errors in Python. Write a definition for a class named Kangaroo with the following methods: 1. An_init_ method that initializes an attribute named pouch_contents to an empty list. 2. A method named put_in_pouch that takes an object of any type and adds it to pouch_contents. 3. A_str method that returns a string representation of the Kangaroo object and the contents of the pouch. Test your code by creating two Kangaroo objects, assigning them to variables named kanga and roo, and then adding roo to the contents of kanga‟s pouch.	**4**
23,24	You will write code that makes Turtles play tag. If you are not familiar with the rules of tag, see http: // en. wikipedia. org/ wiki/ Tag_ (game) . 1. Download http: // thinkpython. com/ code/ Wobbler. py and run it. You should see a TurtleWorld with three Turtles. If you press the Run button, the Turtles wander at random. 2. Read the code and make sure you understand how it works. The Wobbler class inherits from Turtle, which means that the Turtle methods lt, rt, fd and bk work on Wobblers. The step method gets invoked by TurtleWorld. It invokes steer, which turns the Turtle in the desired direction, wobble, which makes a random turn in proportion to the Turtle‟s clumsiness, and move, which moves forward a few pixels, depending on the Turtle‟s speed. 3. Create a file named Tagger.py. Import everything from Wobbler, then define a class named Tagger that inherits from Wobbler. Call make_world passing the Tagger class object as an argument. 4. Add a steer method to Tagger to override the one in Wobbler. As a starting place, write a version that always points the Turtle toward the origin. Hint: use the math function atan2 and the Turtle attributes x, y and heading. 5. Modify steer so that the Turtles stay in bounds. For debugging, you might want to use the Step button, which invokes step once on each Turtle. 6. Modify steer so that each Turtle points toward its nearest neighbor. [Hint: Turtles have an attribute, world, that is a reference to the TurtleWorld they live in, and the TurtleWorld has an attribute, animals, that is a list of all Turtles in the world. 7. Modify steer so the Turtles play tag. You can add methods to Tagger and you can override steer and_ init_ , but you may not modify or override step, wobble or move. Also, steer is allowed to change the heading of the Turtle but not the position.	**4**

	Adjust the rules and your steer method for good quality play; for example, it should be possible for the slow Turtle to tag the faster Turtles eventually.]	
25,26	A vector graphics editor is a program that allows users to draw and edit shapes on the screen and generate output files in vector graphics formats like Postscript and SVG. Write a simple vector graphics editor using Tkinter. At a minimum, it should allow users to draw lines, circles and rectangles, and it should use Canvas.dump to generate a Postscript description of the contents of the Canvas. As a challenge, you could allow users to select and resize items on the Canvas	**4**
27,28	Use Tkinter to write a basic web browser. It should have a Text widget where the user can enter a URL and a Canvas to display the contents of the page. You can use the urllib module to download files (see Exercise 14.6) and the HTMLParser module to parse the HTML tags (see http: // docs. python. org/ 2/ library/ htmlparser. html). At a minimum your browser should handle plain text and hyperlinks. As a challenge you could handle background colors, text formatting tags and images.	**4**
Total Lab hours		**56**

Minor Project(s) – (Only for 2 credit lab)

1. Create a Python project of a Magic 8 Ball which is a toy used for fortune-telling or seeking advice.
   a. Allow the user to input their question.
   b. Show an in progress message.
   c. Create 10/20 responses, and show a random response.
   d. Allow the user to ask another question/advice or quit the game.
2. The “rank” of a word is its position in a list of words sorted by frequency: the most common word has rank 1, the second most common has rank 2, etc. Zipf‟s law describes a relationship between the ranks and frequencies of words in natural languages (http: // en. wikipedia. org/ wiki/ Zipf's_ law). Specifically, it predicts that the frequency, f , of the word with rank r is: f = cr−s where s and c are parameters that depend on the language and the text. If you take the logarithm of both sides of this equation, you get: log f = log c − slog r. So if you plot log f versus log r, you should get a straight line with slope −s and intercept log c. Write a program that reads a text from a file, counts word frequencies, and prints one line for each word, in descending order of frequency, with log f and log r. Use the graphing program of your choice to plot the results and check whether they form a straight line. Can you estimate the value of s?

Suggested Books/Resources:

1. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey , Jeffrey Elkner, 2015
2. Exploring Python, Timothy A. Budd, Mc Graw Hill Education
3. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr
4. Learning Python, Fourth Edition, Mark Lutz, O‟Reilly publication
5. How to Make Mistakes in Python Author: Mike Pirnat
6. Head First Python Paperback – by Paul Barry
7. Link to topics related to course:
   a. Think Python How to Think Like a Computer Scientist
   b. https://greenteapress.com/wp/think-python/
   c. https://www.w3schools.com/python/
   d. https://www.python.org/

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	2	3	1	2	3	3	2	1	1	1	3	**2.1**
CO2	3	3	3	1	2	3	3	3	2	1	1	3	**2.3**
CO3	3	3	3	1	2	3	2	3	2	1	1	3	**2.3**
CO4	3	3	3	3	3	3	3	2	2	1	1	3	**2.5**
CO5	3	3	3	2	2	3	3	3	2	1	1	3	**2.4**
C06	3	3	3	2	3	2	3	2	2	1	1	3	**2.3**
Average	**3**	**2.8**	**3**	**1.7**	**2.3**	**2.8**	**2.8**	**2.5**	**1.8**	**1**	**1**	**3**	

Object-Oriented Systems and Programming Lab

COURSE CODE: 18B17CI371

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: None

Course Objectives:

1. To use object modelling technique to analyze problem requirements, design a solution to the problem and then implement the solution in Object-Oriented Programming Language(s) or database.
2. To strengthen their problem solving ability by applying the characteristics of an object-oriented approach.
3. To strengthen ability to design and represent solutions to problems using UML notations.
4. To introduce object-oriented concepts in C++ and Java.

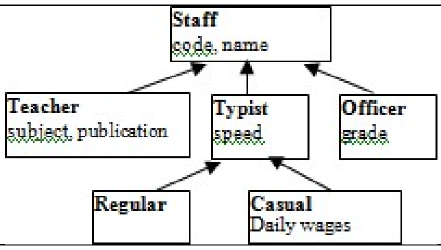
Course Outcomes:

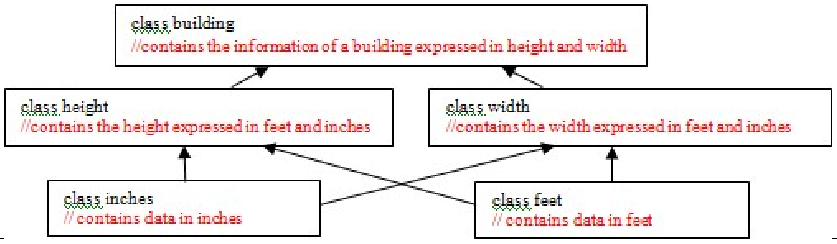
S.No.	Course Outcomes	Level of Attainment
CO1	To learn the concepts of Objects, Classes, Methods, Constructors and Destructors	Usage
CO2	To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions.	Usage
CO3	To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance.	Usage
CO4	To learn the concept of Abstract classes and interfaces	Usage
CO5	To learn the concepts of Operator overloading and conversion function	Usage
CO6	To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file.	Usage
CO7	To learn the Exception Handling: try --catch and finally block, making user-defined exceptions.	Usage
CO8	To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.	Familiarity

List of Experiments

S.No	Description	Hours
1	Create a class called Complex in C++ for performing arithmetic with complex numbers. Use double variables to represent the private data of the class. Provide public member functions for each of the following: *(Define all the member functions outside the class)* a) Enter the values of real and imaginary part. b) Addition of two Complex numbers: (The real parts are added together and the imaginary parts are added together. c) Subtraction of two Complex numbers. (The real part of the right operand is subtracted from the real part of the left operand and the imaginary part of the right operand is subtracted from the imaginary part of the left operand. ***d)*** Printing Complex numbers in the form: **a + b *i***	**2**
2	Define a class in C++ to represent a bank account. Include the following members:	**2**

	Data members: a) Name of the depositor b) Account number c) Type of account d) Balance amount in the account ***Member functions:*** *(Define all the member functions inside the class)* a) To assign initial values b) To deposit an amount c) To withdraw an amount after checking the balance d) To display name and balance Write a main function to create a database of 10 customers	
3	a) Employees have a number, date of birth (dd-mm-yyyy), rank, and salary. When an employee is first recruited then all these are given values of 0. Upon confirmation, the actual values of these are entered for the employee. Their rank can be incremented by 1 and when this happens an employee gets an increment of 25%. Write a C++ class for Employee. b) Students are registered in a University. When students are created then they are given default values (zeroes or blanks) for roll_number, department, year, and semester of study. At registration time, the values of these attributes of student are updated with the proper values. Students can be promoted and their departments can be changed. Write a C++ class for Student.	**2**
4	Users of the computer have profile which consists of Name, Password, and Access Rights. Access rights take on values X, R, W, and ALL. It is possible to have default values for Password and Access rights which are the first three letters of the Name and ALL respectively. Users can change their password and access rights. Write a class User in C++ and create a user named Rajesh.	**2**
5	Define two classes Distance1 and Distance2 in C++. Distance1 stores distance in miles and Distance2 in kmeters & meters. Write a program that reads values of the class objects and adds one object of Distance1 with the object of Distance2 class. The display should be in the format of miles or kmeters & meters depending on the type of object (Distance1 or Distance2) being used to invoke the function. (Hint: Make use of friend function).	**2**
6	Implement a singleton class in C++. A class whose number of instances that can be instantiated is limited to one is called a singleton class. (Hint: make use of static members).	**2**
7	Imagine a publishing company that markets both books and audio- cassette version of its works. Create a class Publication in C++ that stores the title (a string) and price (type float) of a publication. From this class derive two classes: Book, which adds a page count and Tape, which adds playing time in minutes. These classes should have getdata() function to get its data from the user and the putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in their data with getdata() and displaying the data with putdata().	**2**
8	Implement the class hierarchy as shown in the following figure (using C++).	**2**

	 Staff code, name Teacher subject, publication Typist speed Officer grade Regular Casual Daily wages	
19	The database created in the experiment 8 does not include educational information of the staff. It has been decided to add this information to teacher and officers (not for typists) which will help the management in decision making with regard to training, promotion, etc. Add another data class called Education that holds two pieces of education information, namely, highest qualification in general education and highest professional qualification. The class should be inherited by the class Teacher and Officer. Modify the program of above exercise to incorporate these additions. Include overloaded constructors in all above classes.	**2**
10	Implement the class hierarchy shown in the following figure, using C++. Define appropriate member functions (including constructors and destructors) to convert feet class object into inches class object and vice versa. Also the objects of the feet and inches constructors should construct their objects using the constructors of the height and width constructor which in turn call building constructor.	**2**

	 class building
	//contains the information of a building expressed in height and width
	class height
	//contains the height expressed in feet and inches
	class width
	//contains the width expressed in feet and inches
	class inches
	// contains data in inches
	class feet
	// contains data in feet	

11	Define a class Directory with members: name and phone number. Use the class object to store each set of data into a text file “phone.txt”. The names contain only one word and the names and telephone numbers are separated by white spaces. Write a C++ program to read the file and output the list in two columns, such as: John 23456 Ahmed 9876	**2**
12	Write an interactive, menu-driven program that will access the file created in the experiment 11 and implement the following tasks: Determine the telephone number of the specified person. b) Determine the name if a telephone number is known. Update the telephone number, whenever there is a change	**2**
13	Define a class Queue in C++ that contains elements of type integer. Define two operators on Queue, „+‟ to insert an element in it and „–„ to remove and element from it. Use the friend function approach first and then the one without friends	**2**
14	A programmer wants to manipulate arrays. Two arrays are equal if (a) they have the same dimension, (b) are of the same size, and (c) contain identical values in their corresponding elements. Comparison is done using the operator „= =‟ which returns true or false. Also, arrays can be copied to one another using the operator „=‟. Implement the foregoing using the friend function approach first and then the one without friends. Which one is preferable and why?	**2**
15	An istream class overloads the >> operator for the standard types [int, long, double, float, char]. For example, the statement cin >> x ; calls the appropriate >> operator function for the istream class defined in iostream.h and uses it to direct this input	**2**

	stream into the memory location represented by the variable x. Similarly, the ostream class overloads the << operator, which allows the statement cout << x to send the value of x to ostream cout for output. Overload these operators to enter the object‟s data members through input operator >> and display the values of these members by using the output operator <<.	
16	Define a class Distance in C++ with data members: kmeter and meter. Define conversion function to convert Distance object into distance in miles (float type). Also make use of constructor to convert distance in miles (float type) into object of Distance class.	**2**
17	Define two classes Polar and Rectangle (using C++) to represent points in polar and rectangle systems. Use the conversion routines to convert from one system to the other.	**2**
18	Define a function template in C++ to sort an array of elements of int type, string type, float type and user-defined type Distance with data members: kmeter & meter.	**2**
19	Define a template class LinkedList in C++ with the following member functions: createList(), displayList(), insertElement(), and deleteElement(); In the main() function invoke above functions for the list of integer numbers, floating numbers and complex numbers.	**2**
20	Define a class Employee in Java with members: name, age and salary and methods: enterData() to enter the record of the employee through keyboard: displayData() to display the details of the employee. Define two user defined exceptions that are thrown when) The name entered has numeric character(s).) The age of the person is less than 18 or greater than 60.	**2**
21	Write a program in Java that demonstrates handling of exceptions in inheritance tree. For example, create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge() where the input age <0. In Son class, implement a constructor that uses both father and son‟s age and throws an exception if son‟s age is >= father‟s age.	**2**
22	Define a class Person in the package MyPackage. The class has data members as: name, age, address and Methods to enter the data through keyboard and display them. Make use of overloaded constructors in the class. Now, import the above class and inherit the class Employee from this. The sub-class should have overloaded functions and also call to the base class constructors. In the main class, define a database of 5 objects and display them.	**2**
23	Define an abstract class Shape in Java, with two abstract functions: enterData() and displayArea(). Define two classes Circle and Rectangle inside a new file that inherit the class Shape and implement the abstract functions in their own way. Illustrate the concept of dynamic binding in this program.	**2**
24	Find the source code for the Vector class in the Java source code library that comes with all Java distributions. Copy this code and make a special version called intVector that holds only integers. Consider what it would take to make a special version of Vector for all the primitive types. Now consider what happens if you want to make a linked list class that works with all the primitive types. If parameterized types are ever implemented in Java, they will provide a way to do this work for you automatically.	**2**

25	Create a class MyString in Java containing a String object so that you initialize in the constructor using the constructor"s argument. Add a toString() method and a method concatenate() that appends a String object to your internal string. Implement clone() in MyString. Create two static methods that each take a myString x handle as an argument and call x.concatenate("test"), but in the second method call clone() first. Test the two methods and show the difference effects.	
26	**Design the USE-CASE diagram for the following:** A description of the behavior of an automated telling machine (ATM) is given below: A user begins a transaction at the ATM by entering a bank card. If the card is readable by the machine the user is prompted to enter their personal identification number (PIN). Once this number has been entered, a menu is presented to the user containing the following options: show account balance, withdrawal with receipt and withdrawal without receipt. If the user selects one of the withdrawal options, they are prompted to enter an amount of money to withdraw; the amount entered must be a multiple of 10. The user"s PIN is validated when the ATM sends the details of the transaction to the bank"s remote computer. If the PIN is invalid, the user is given the option of re-entering it and the selected transaction is retired. This is repeated if the new PIN is also invalid. Once three invalid PINs have been entered, the transaction processing depends on the transaction type selected. For a show balance transaction, the balance is displayed on the screen and after they have confirmed this, the user is returned to the transaction menu. A withdrawal transaction may fail if the user has executed the amount of money that can be withdrawn from the account; in this case an error message is displayed and, after confirmation, the user is returned to the transaction menu. Otherwise, the user"s card is returned and the money is issued, followed by the receipt if required. At any point where user input, other than a simple confirmation, is required, a cancel option is provided. If this is selected, the user"s card is returned and their interaction with the ATM terminates.	**2**
27	**Design the Collaboration and Sequence diagram for the following:** Many word processors, graphical editors and similar tools provide cut, copy and paste operations by means of some kind of a clipboard facility. At run-time suppose that such a system consists of an instance of an Editor class linked to a number of instances of a Element class. Elements are the items manipulated by a tool, such as words or shapes. Some of the elements may have been selected by the user. The editor is also linked to an instance of a Clipboard class, which in turn maintains links to all the elements that have been placed in the clipboard. An element cannot simultaneously be displayed by the editor and on the clipboard. Draw a collaboration showing a configuration where there are several elements displayed by the editor and one element on the clipboard. Assume that the selected elements are identified by an additional link	**2**
	from the editor. Draw a sequence diagram showing what happens when the editor receives a cut message from a client. The effect of this is that all the currently selected elements are moved to the clipboard.	

	Draw a sequence diagram showing what happens when the editor receives a paste message from a client. The effect of this is that all the elements on the clipboard are moved back to the editor. Draw a sequence diagram showing what happens when the editor receives a copy message from a client. The effect of this is that all the currently selected elements are copied to the clipboard. Assume that elements implement a clone operation, which returns a exact copy of the element. Draw equivalent collaboration diagrams for each of your answers.	
28	**Draw a state diagram expressing these facts about the display of windows.** A window in a window management system on a computer can be displayed in one of the three states: maximized, where it takes up the entire screen; normal, where it is displayed as a small icon. When a window is opened, it will be displayed as a normal window, unless minimize on use has been selected, in which case it will be displayed as an icon. A normal window and an icon can be maximized; a maximized window and a normal window can be minimized or reduced to an icon. Maximized windows can be restored to their normal size and icons can be restored to the size they had before they were minimized. Icons and normal windows can be moved as normal windows can also be resized. No matter how it is displayed, a window can always be closed.	**2**
Total Lab hours		**56**

Suggested Books/Resources:

1. Lafore R., Object oriented programming in C++, Waite Group
2. Java 2: The Complete Reference, Fifth Edition -- by Herbert Schildt
3. Satzinger, Jackson, Burd, Object-Oriented Analysis & Design
4. Stroustrap B., The C++ Programming Language, Addison Wesley
5. Bruce Eckel, Thinking in C++
6. Bruce Eckel, Thinking in Java

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO2	3	3	3	2	1	1	2	3	2	3	3	3	**2.4**
CO3	3	3	2	2	2	1	2	2	2	3	3	3	**2.3**
CO4	3	3	3	2	2	1	2	3	2	3	3	3	**2.5**
CO5	3	3	3	3	2	1	2	3	2	1	3	3	**2.4**
C06	3	3	3	3	2	1	2	3	2	2	3	3	**2.5**
CO7	3	3	2	2	2	3	3	2	2	3	3	3	**2.6**
CO8	3	3	3	3	3	2	3	3	2	1	3	1	**2.5**
Average	**3**	**3**	**2.8**	**2.5**	**2**	**1.5**	**2.4**	**2.8**	**2**	**2.4**	**3**	**2.8**	

Database Management Systems Lab

COURSE CODE: 18B17CI373

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: None

Course Objectives:

1. Develop the ability to design, implement and manipulate databases.
2. Introduce students to build database management systems.
3. Apply DBMS concepts to various examples and real life applications.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO1	Design and implement a database schema	Usage
CO2	Design different views of tables for different users and to apply embedded and nested queries	Usage
CO3	Understand the use of structured query language and its syntax , transactions, database recovery and techniques for query optimization	Familiarity
CO4	Understand , analyze and apply common SQL statements including DDL , DML , DCL statements to perform different operations	Assessment & Usage
CO5	Develop application programs using PL/SQL	Usage
CO6	Design and implement a project using embedded SQL and GUI	Usage

List of Experiments

S.No	Description	Hours
1	To implement Data Definition language commands Create database/table, alter, drop, truncate	**2**
2	To implement Constraints as a part of Data Definition language Primary key, Foreign Key, Check, Unique	**2**
3	To implement Constraints as a part of Data Definition language Null, Not null, Default, Enable Constraints, Disable Constraints, Drop Constraints	**2**
4	To implement Data Manipulation Language Commands Insert, Select, Update, Delete	**2**
5	To implement Data Control Language, Transfer Control Language Commands commit, rollback, save point, grant, revoke	**2**
6	To practice in Built Functions Date functions, numerical functions, character functions, conversion functions, group functions, count functions etc.	**2**
7	To practice group by, having clause and special operators such as between, like, in etc.	**2**
8	To practice Nested Queries	**2**
9	To practice Nested Queries and Join Queries Inner join, Left join, Right join, Full join	**2**
10	To implement Set Operators Union, Intersect, Minus	**2**
11	To implement Views	**2**

12	To implement and practice PL/SQL control structure If, if then else, else if, nested if	**2**
13	To implement and practice PL/SQL control structure For loop, while loop	**2**
14	To implement and practice PL/SQL procedures	**2**
15	To implement and practice PL/SQL functions	**2**
16	To implement triggers	**2**
17	To study about various Visual Basic (front end) tools	**2**
18	To design and implement forms using visual basic	**2**
19	To design and implement a menu design using Visual Basic	**2**
20	To implement report generation using VB.	**4**
21	To create a database for payroll processing system using SQL	**4**
22	Implement the above created database using VB.	**4**
23,24,25	**Minor Projects – (Only for 2 credit lab)** Banking System University System Company System Hospital Management System Passport Automation System	**6**
Total Lab hours		**56**

Suggested Books/Resources:

1. "Fundamentals of Database Systems" Elmasri, Navathe, Pearson Education.
2. “Database system concepts” Henry F Korth, Abraham Silberschatz, S. Sudurshan, McGraw-Hill
3. “Database Systems: A Practical Approach to design, Implementation and Management”. Thomas Connolly, Carolyn Begg; Third Edition, Pearson Education.
4. Bipin C Desai, ?An Introduction to Database Systems?, Galgotia. Publications Pvt Limited, 2001
5. “An Introduction to Database Systems”, C.J.Date, Pearson Education.
6. “A first course in Database Systems”, Jeffrey D. Ullman, Jennifer Windon, Pearson, Education.
7. Orcacle manual
8. Link to topics related to course:
   a. https://www.youtube.com/watch?v=EUzsy3W4I0g&list=PL9426FE14B809CC41
   b. https://www.w3schools.com/sql/
   c. https://www.codementor.io/collections/learn-sql-bwclmlodl

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	2	2	2	2	2	1	2	3	3	1	**2.2**
CO2	3	1	3	2	2	2	3	2	2	3	2	2	**2.3**
CO3	3	3	2	1	1	3	3	3	3	3	1	1	**2.3**
CO4	3	2	2	2	2	2	2	2	1	2	1	1	**1.8**
CO5	3	3	2	1	2	3	3	2	1	2	2	1	**2.1**
C06	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
Average	**3**	**2.5**	**2.2**	**1.5**	**1.7**	**2.5**	**2.7**	**1.8**	**2**	**2.7**	**2**	**1.2**	

IT Workshop Lab

COURSE CODE: 18B17CI372

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: None

Course Objectives:

1. To introduce the students with the basic features of Matlab for problem solving
2. To introduce the students about the Mathematical functions like matrix generation and Plotting with multiple data sets, line styles and colors.
3. To introduce the students about the Array operations and solving Linear equations in Matlab.
4. To introduce the students about the control flow and operators using if-end structures and loops.
5. To introduce the students about the writing M-file scripts and Debugging M-files

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO1	Writing fundamental programs in Matlab, creating variables and mathematical functions	Familiarity
CO2	Programming the fundamentals concepts of basic Plotting consisting of simple and multiple data sets in one plot	Usage
CO3	Understand how to program matrix operations, array operations and how to solve the system of linear equations	Assessment
CO4	Understand how to program M-file scripts , M- file functions, Input –output Arguments	Assessment
CO5	Program control flow operators, loops, flow structures and debugging M-files	Assessment

List of Experiments

S.No	**Description**	**Hours**
1	1. Create variable, pounds, to store a weight in pounds. Convert this to kilograms and assign the result to variable kilos. The conversion factor is 1 kilogram = 2.2 pounds. 2. The combined resistance RT of three resistors R1, R2, and R3 in parallel is given by	**2**
	$RT = \frac{1}{\frac{1}{R_1}+\frac{1}{R_2}+\frac{1}{R_3}}$ Create variables for the three resistors and store values in each, and then calculate the combined resistance.	
2	Write a MATLAB program to calculate the following expression and round the answers to the nearest integer. a) z = 5x2 + y2 where x=2, y=4 b) z = 4cos(x)+ j6sin(x) where x=π/4 c) z = 3sin(x)+4cos(x)+3ey where x=π/3 , y=2 d) y=sin(x) / x where 0 ≤ x ≤ 2π 2. Solve the following system x + y - 2z =	**2**

	3 2x + y = 7 x + y - z = 4	
3	1. Write a program for three bits parity generator using even-parity bit. 2. Write a program to convert a three bits binary number into its equivalent gray code. 3. if q=[1 5 6 8 3 2 4 5 9 10 1],x=[3 5 7 8 3 1 2 4 11 5 9], then: a) find elements of (q) that are greater than 4. b) find elements of (q) that are equal to those in (x). c) find elements of (x) that are less than or equal to 7. 4. If x=[10 3 ; 9 15], y=[10 0; 9 3], z=[-1 0; -3 2], what is the output of the following statements: a) v = x > y b) w = z >= y c) u = ~z & y d) t = x & y < z	**2**
4	1.Plot sin(x) on the interval [-pi,pi] using spacing 0.5, 0.1 and 0.01 between the points where you will sample the function. (This will change the resolution). Experiment with the hold on command. 2. Attach labels to the axis of the previous plot and give a title to the graph. 3. Plot 5 cos(x2+1) on [-2pi,2pi]. Note that the squaring operation will require you to use the dot . in order for the squaring operation to act on each element individually. However, the addition operation (+) automatically acts on elements individually.	**2**
5	1. Type x=[1 2 3] y =[4 5 6] a=2 x+y x-y a*x and observe what happens. If want to apply an operation such as squaring each element in a matrix we have to use a dot . before the operation we wish to apply. Type the following commands in MATLAB. x=1:10 x.^2 A=[1 2 3 ; 4 5 6 ; 7 8 9] A.^2 A^2 and observe the result. The dot allows us to do operations element wise.	**2**

	All built-in functions such as sin, cos, exp and so on automatically act elementwise on a matrix. Type y=[0 1/4 1/2 3/4 1] y=pi*y sin(y) and observe the result. 2. Create a array x with 10 elements x=[1 2 3 4 5 6 7 8 9 10] We can also create this vector by typing x=1:10. The vector (1 1.1 1.2 1.3 1.4 1.5) can be created by typing x=[1 1.1 1.2 1.3 1.4 1.5] or by typing x=1:0.1:1.5. Matrices can be created according to the following example. The matrix A= $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ is created by typing A=[1 2 3 ; 4 5 6; 7 8 9], i.e., rows are separated with semi-colons. If we want to use a specific element in a vector or a matrix, study the following example: Example: x=[10 20 30] A=[1 2 3 ; 4 5 6 ; 7 8 9] x(2) A(3,1) Here we extracted the second element of the vector by typing the variable and the position within parentheses. The same principle holds for matrices; the first number specifies the row of the matrix, and the second number specifies the column of the matrix. Note that in MATLAB the first index of a vector or matrix starts at 1, not 0 as is common with other programming languages.	
6	1. A Pythagorean triple is a set of positive integers (a,b,c) such that a2 + b2 = c2. Write a function ispythag that will receive three positive integers (a, b, c in that order) and will return 1 for true if they form a Pythagorean triple, or 0 for false if not. 2. Whether a storm is a tropical depression, tropical storm, or hurricane is determined by the average sustained wind speed. In miles per hour, a storm is a tropical depression if the winds are less than 38 mph. It is a tropical storm if the winds are between 39 and 73 mph, and it is a hurricane if the wind speeds are > = 74 mph. Write a script that will prompt the user for the wind speed of the storm, and will print which type of storm it is.	**2**
7	1. Write a script that will prompt the user for N integers, and then write the positive numbers (>= 0) to an ASCII file called pos.dat and the negative numbers to an ASCII file called neg.dat. Error-check to make sure that the user enters N integers. 2. Write a script that will continue prompting the user for positive numbers, and storing them in a vector variable, until the user types a negative number. 3. Write a script that will use the menu function to present the user with choices for functions fix, floor, and ceil. Error-check by looping to display the menu until the user pushes one of the buttons (an error could occur if the user clicks the X on the menu box rather than pushing one of the buttons). Then, generate a random number and print the result of the user‟s function choice of that number (e.g., fix(5)).	2

8	1. Write a "currency exchange program" similar to the one in Example 1 which can handle two different exchange rates, exchange_rate1 = 0.5 and exchange_rate2 = 0.25. Design the program to first ask for the amount in dollars and then ask the user which rate (represented by the numbers 1 and 2 respectively) he/she wants. Let the program return the amount in the requested foreign currency. 2. Write a program that approximates PI by computing the sum . The more terms you keep in the summation, the more accurate your answer will be. (In fact, the series converges to PI as m goes to infinity.) See how many terms you need to approximate PI with 5 decimals. (Note: This is by no means the most efficient way to approximate PI, but the formula is quite beautiful...) 3. Use the sum given in Exercise 2 to approximate PI using 10, 100, 1000, 10000 and 100000 terms. For each of these numbers, compute the error of the approximation. Plot the error as a function of the number of terms used in the sum.	2
9	1. In Europe daylight time starts on the last Sunday of March and ends on the last Sunday of October. Write a function that determines whether a given daynumber is in the summertime period or in the wintertime period of the Daylight Saving Time 2. Write a function that will receive the radius of a circle and will print both the radius and diameter of the circle in a sentence format. This function will not return any value; it simply prints	**2**
10	1. Write a program to calculate and print the area and circumference of a circle. There should be one script and three functions to accomplish this (one that prompts for the radius, one that calculates the area and circumference, and one that prints). 2. The lump sum S to be paid when interest on a loan is compounded annually is given by S = P(1 + i)n, where P is the principal invested, i is the interest rate, and n is the number of years. Write a program that will plot the amount S as it increases through the years from 1 to n. The main script will call a function to prompt the user for the number of years (and error-check to make sure that the user enters a positive integer). The script will then call a function that will plot S for years 1 through n. It will use 0.05 for the interest rate and $10,000 for P.	**2**

11	1. Write a script that will prompt the user for a temperature in degrees Celsius, and then an F for Fahrenheit or K for Kelvin. The script will print the corresponding temperature in the scale specified by the user. For example, the output might look like this: Enter the temp in degrees C: 29.3 Do you want F or K? F The temp in degrees F is 84.7 The format of the output should be exactly as specified here. The conversions are: F = 9/5 *C+32 K = C+273.15 2. Write a function to calculate the volume and surface area of a hollow cylinder. It receives as input arguments the radius of the cylinder base and the height of the cylinder. The volume is given by Π r2 h, and the surface area is 2 Π r h. 3. Hurricanes are categorized based on the winds. The following table shows the category number for hurricanes with varying wind ranges and what the storm surge is (in feet above normal).	**2**
	1 74–95 4–5 2 96–110 6–8 3 111–130 9–12 $\pi \approx 4\sum_{k=0}^{m}\frac{(-1)^k}{2k+1}$ 4 131–155 13–18 5 >155 >18 Write a function that will receive as an input argument the wind speed, and will return the category number and the minimum value of the typical storm surge.	

2	1. Write a function called geomser that will receive values of r and n, and will calculate and return the sum of the geometric series: 1 + r + r2 + r3 + r4 + ... + rn The following examples of calls to this function illustrate what the result should be: >> geomser(1,5) ans = 6 >> disp(geomser(2,4)) 31 2. A sound engineer has recorded a sound signal from a microphone. The sound signal was sampled, meaning that values at discrete intervals were recorded (rather than a continuous sound signal). The units of each data sample are volts. The microphone was not on at all times, however, so that data samples below a certain threshold are considered to be data values that were samples when the microphone was not on, and therefore not valid data samples. The sound engineer would like to know the average voltage of the sound signal. Write a script that will ask the user for the threshold and the number of data samples, and then for the individual data samples. The program will then print the average and a count of the valid data samples, or an error message if there were no valid data samples. An example of what the input and output would look like in the Command Window is shown: Please enter the threshold below which samples will be considered to be invalid: 3.0 Please enter the number of data samples to be entered: 7 Please enter a data sample: 0.4 Please enter a data sample: 5.5 Please enter a data sample: 5.0 Please enter a data sample: 2.1 Please enter a data sample: 6.2 Please enter a data sample: 0.3 Please enter a data sample: 5.4 The average of the 4 valid data samples is 5.53 volts	2
13	1. Create a vector of five random integers, each in the range from –10 to 10. Perform each of the following two ways: using built-in functions, and also using loops (with if statements if necessary): Subtract 3 from each element. Count how many are positive. Get the absolute value of each element. Find the maximum. 2. Create a 3 × 5 matrix. Perform each of the following two ways: using built-in functions, and also using loops (with if statements if necessary): Find the maximum value in each column. Find the maximum value in each row. Find the maximum value in the entire matrix. 3. Write a script that will print the following multiplication table: 1 2 4 3 6 9 4 8 12 16 4. 5 10 15 20 25	2

14	1. Biomedical engineers are developing an insulin pump for diabetics. To do this, it is important to understand how insulin is cleared from the body after a meal. The concentration of insulin at any time t is described by the equation C = C0 e−30t/m where C0 is the initial concentration of insulin, t is the time in minutes, and m is the mass of the person in kg. Write a script that will graphically show how the weight of the person influences the time for insulin to be cleared from the body. It will show in a 2 × 1 subplot the concentration of insulin for two subjects, one who weighs 120 pounds, and one who weighs 300 pounds. For both, the time should increment from 0 to 4 minutes in steps of 0.1 minute, and the initial concentration should be 85. The concentration over time will be shown in each subplot, and the weight of the person should be in the title. The conversion factor is 1 pound = 0.4536 kg. In order to better compare, use consistent axes for both plots. 2. Sales (in millions) from two different divisions of a company for the four quarters of 2006 are stored in vector variables, for example, div1 = [4.2 3.8 3.7 3.8]; div2 = [2.5 2.7 3.1 3.3]; Using subplot, show side-by-side the sales figures for the two divisions. What kind of graph shows this in the best way? Why? In one graph, compare the two divisions. What kind of graph shows this in the best way? Why?	**2**
15	1. For the following matrices A, B, and C: A B C 1 4 2 1 3 3 2 5 3 2 1 5 6 4 1 2 3 6 0 Which are symmetric? For all square matrices, give their trace. Give the result of 3*A. Give the result of A*C. Are there any other matrix multiplications that can be performed? If so, list them. 2. Given the following matrices: A B C 3 2 1 2 1 0 0 0 5 2 1 0 1 0 1 0 3 3 0 0 1 Perform the following MATLAB operations, if they can be done. If not, explain why. A * B B * A I +A A .* I trace(A)	**2**
16	1. Write a function issquare that will receive an array argument, and will return 1 for true if it is a square matrix, or 0 for false if it is not. 2. Write a function mydiag that will receive an array argument, and will return a vector consisting of the main diagonal (without using the built- in diag function).	**2**

17	2. Write a function that will receive a square matrix as an input argument, and will return a row vector containing the diagonal of the matrix. If the function is called with a vector of two variables on the left-hand side of the assignment, the function will also return the trace of the matrix. (Note: It will return the trace only if there is two variables on the left-hand side of the assignment.) You may assume that the matrix is square. The function must preallocate the diagonal vector to the correct size. 2. Write a function randdiag that will return an n x n diagonal matrix, with random integers each in the range from low to high on the diagonal. Three arguments are passed to the function: the value of n, low, and high, in that order.	**2**
18	1. Write a function to receive a matrix and return its transpose (for more programming practice, do not use the built-in operator for the transpose). 2. We have already seen the zeros function, which returns a matrix of all 0‟s. Similarly, there is a function ones that returns a matrix of all 1‟s. Note: No, there aren‟t functions called twos, threes, and such (just ones and zeros!). However, write a fives function that will receive two arguments for the number of rows and columns and will return a matrix with that size of all 5‟s.	**2**
19	1. The function magic(n) returns an n n magic matrix, which is a matrix for which the sum of all rows, columns, and the diagonal are the same. Investigate this built-in function. 2. The function pascal(n) returns an n n matrix made from Pascal‟s triangle. Investigate this built-in function, and then write your own.	**2**
20	1. For the following 2 x 2 system of equations: 3x1 + 2x2 = 4x1 = 2 Write this in matrix form. Using the method for 2 x 2 systems, find the determinant D. Use D to find the inverse of A. Use the Gauss elimination method to find the solution. Use the Gauss-Jordan method to solve. Check your work in MATLAB. 2. For the following set of equations: 2x1 + 2x2 + x3 = 2x 2 + 2x3 = 1 x1 + x2 + 3x3 = 3 Put this in the augmented matrix [A|b]. Solve using Gauss. Solve using Gauss-Jordan. In MATLAB, create the matrix A and vector b. Find the inverse and determinant of A. Solve for x.	**2**
21	1. Solve the simultaneous equations x – y = 2 and x2 + y = 0 using solve. Plot the corresponding functions, y = x – 2 and y = –x2, on the same graph with an x range from –5 to 5. 2. For the following set of equations: 2x1 + 2x2 + x3 = 2 x 2 + 2x3 = 1	**2**

	x1 + x2 + 3x3 = 3 In MATLAB, create the coefficient matrix A and vector b. Solve for x using the inverse, using the built-in function. Create the augmented matrix [A|b] and solve using the rref function. Write this in symbolic form and solve using the solve function. From the symbolic solution, create a vector of the numerical (double) equivalents	
22	1. Rewrite the following system of equations in matrix form: 4x1 − x2 + 3x4 = 10 −2x1 + 3x2 + x3 − 5x4 = −3 x1 + x2 − x3 + 2x4 = 2 3x1 + 2x2 − 4x3 = 4 Set it up in MATLAB and use any method to solve. 2. For the following 2 2 system of equations: −3x1 + x2 = −4 −6x1 + 2x2 = 4 In MATLAB, rewrite the equations as equations of straight lines and plot them to find the intersection. Solve for one of the unknowns and then substitute into the other equation to solve for the other unknown. Find the determinant D. How many solutions are there? One? None? Infinite?	**2**
23	1. For the following 2 x 2 system of equations: −3x1 + x2 = 2 −6x1 + 2x2 = 4 Rewrite the equations as equations of straight lines and plot them to find the intersection. Solve for one of the unknowns and then substitute into the other equation to solve for the other unknown. Find the determinant D. How many solutions are there? One? None? Infinite? 2. Write a function to return the determinant of a 2 x 2 matrix. 3. Write a function to return the inverse of a 2 x 2 matrix.	**2**
24	1. Write a script that will do the following. Create two vectors with 20 random integers in each; in one the integers should range from 1 to 5, and in the other from 1 to 500. For each vector, would you expect the mean and median to be approximately the same? Would you expect the standard deviation of the two vectors to be approximately the same? Answer these questions, and then use the built-in functions to find the minimum, maximum, mean, median, standard deviation, and mode of each. Do a histogram for each in a subplot. Run the script a few times to see the variations. 2. Write a function that will return the mean of the values in a vector, not including the minimum and maximum values. Assume that the values in the vector are unique. It is OK to use the built-in mean function. To test this, create a vector of 10 random integers, each in the range from 0 to 50, and pass this vector to the function.	**2**

25	1. A student missed one of four exams in a course, and the professor decided to use the average of the other three grades for the missed exam grade. Which would be better for the student: the mean or the median if the three recorded grades were 99, 88, and 95? What if the grades were 99, 70, and 77? 2. A weighted mean is used when there are varying weights for the data values. For a data set given by x = {x1, x2, x3, x4, . . . , xn} and corresponding weights for each xi, w = {w1, w2, w3, w4, . . . , wn}. Write a function that will receive two vectors as input arguments: one for the data values and one for the weights, and will return the weighted mean.	**2**
26	1. DNA is a double-stranded helical polymer that contains basic genetic information in the form of patterns of nucleotide bases. The patterns of the base molecules A, T, C, and G encode the genetic information. Construct a cell array to store some DNA sequences as strings, such as TACGGCAT ACCGTAC and then sort these alphabetically. Next, construct a matrix to store some DNA sequences of the same length and then sort them alphabetically. 2. Write a function that will receive two arguments: a vector and a character (either „a‟ or „d‟) and will sort the vector in the order specified by the character (ascending or descending).	**2**
27	1. Write a function matsort to sort all the values in a matrix (decide whether the sorted values are stored by row or by column). It will receive one matrix argument and return a sorted matrix. Do this without loops, using the built-in functions sort and reshape. For example: >> mat mat = 4 5 2 1 3 6 7 8 4 9 1 5 >> matsort(mat) ans = 1 4 6 1 4 7 2 5 8 3 5 9 2. Write a function that will receive two arguments: a vector and a character (either „a‟ or„d‟) and will sort the vector in the order specified by the character (ascending or descending).	**2**

8	1. Find the roots of the equation f(x) = 0 for the following function. Also, create x and y vectors and plot this function in the range from –3 to 3 in order to visualize the solution. f(x) = 3x2 - 2x − 5 2. Evaluate the polynomial expression 3x3 + 4x2 + 2x − 2 at x = 4, x = 6, and x = 8. 3. Sometimes the roots of polynomial equations are complex numbers. For example, create the polynomial row vector variable pol: >> pol = [3 6 5]; Use the roots function to find the roots. Also, use ezplot(poly2sym(pol)) to see a plot. Then, change the last number in pol from 5 to –7 and again find the roots and view the plot.	**2**
Total Lab hours		**56**

Suggested Books/Resources:

1. Stormy Attaway, Matlab: a Practical Introduction to Programming and Problem Solving, Elsevier
2. Essentials of MATLAB Programming Stephen J. Chapman , 2005
3. MATLAB for Engineers Holly Moore, 2007
4. MATLAB Programming for Engineers Stephen J. Chapman , 1999
5. Matlab, An Introduction With Applications Amos Gilat , 2003
6. MATLAB Guide 2000
7. https://nptel.ac.in/courses/103106118/
8. Link to topics related to course:
   i. https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf
   ii. https://www.math.utah.edu/~wright/misc/matlab/matlabintro.html
   iii. https://web.stanford.edu/class/ee254/software/using_ml.pdf

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	3	1	1	2	1	2	1	2	2	2	**1.9**
CO2	3	3	3	1	3	2	1	2	1	2	3	2	**2.2**
CO3	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO4	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO5	3	3	3	3	2	1	2	2	2	2	2	2	**2.3**
Average	**3**	**3**	**3**	**2.2**	**2**	**1.4**	**1.6**	**2**	**1.6**	**2**	**2.6**	**2**	

Discrete Computational Mathematics

COURSE CODE: 18B11CI414

COURSE CREDITS: 3

CORE/ELECTIVE: COR

L-T-P: 3-0-0

Pre-requisite: Basic Mathematics Algebra

Course Objectives

1. To simplify and evaluate any logical expression and to express logical statements in terms of logical connectives, predicates and quantifiers.
2. Use of various set operations, relations and functions concept to solve applied problems.
3. To solve counting problems using elementary counting techniques.
4. To learn and perform various graphs and trees terminologies, traversals & their applications.
5. Problem solving using recursion and recurrence relations by analyzing algorithms.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Students will be able to express logical statements in terms of logical connectives, predicates and quantifiers.	Familiarity
CO-2	Students will be able to apply various proving techniques such as direct, indirect proofs, mathematical induction, etc.	Assessment
CO-3	They will learn basic set operations along with relations & functions with their types and usage.	Familiarity
CO-4	They will be familiar with graph & tree terminologies along with their various applications in computer science.	Familiarity
CO-5	Students will be able to solve counting problems using permutation, combinations techniques.	Assessment
CO-6	They will learn about algebraic structures such as group, abelian group, rings, integral domain, fields, etc	Familiarity
CO-7	Students will be able to analyze and solve various algorithms using recurrence relation methods	Assessment

Course Contents:

Unit	Contents	Lectures required
1	Introduction and Applications of Discrete Mathematics, Mathematical Logic: Propositional & Predicate; Quantifiers, Proving Techniques: Direct Proof, Contra positive, Contradiction, Principle of Mathematical Induction; Pigeonhole Principle	6
2	Sets, Types of Sets, Various set operations, Venn Diagrams, Identities in sets, Principle of Inclusion & Exclusion	3
3	Relations: Types & Representation; Properties of Binary Relations, Equivalence Relations, Partial Ordering Relations, Partitions. Functions, Types of Functions, inverse of function, composition of functions.	5
4	Graph, Graph Terminologies, Types of Graphs, Paths & Circuits, Euler & Hamiltonian Graphs, Planar Graphs, Graph Traversals: Breadth First Search & Depth First Search, Shortest Path Algorithms.	7

5	Trees, Tree Terminologies, Types of Trees: General, Binary, Strictly Binary, Full & Complete Binary Tree; Tree Traversals, Binary Search Tree, AVL Trees.	7
6	Basic Counting Techniques, The Sum and Product Rule, Permutations, Combinations, Generation of Permutations and Combinations	4
7	Properties of Algebric Structures, Semigroups, Monoids, Groups, Abelian Groups, Subgroups, Homomorphism & Isomorphism of Groups, Rings, its characteristics & its types, Integral Domain & Fields.	6
8	Recurrence Relations, Linear Recurrence Relations with constant coefficients (homogeneous & non-homogeneous) with their solving techniques.	4
Total lectures		**42**

Suggested Text Book(s):

a. C.L. Liu & D.P. Mohapatra, “Elements of Discrete Mathematics: A Computer Oriented Approach”, 4th Edition, TMH

b. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, 7th Edition, TMH

Suggested Reference Book(s):

1. B. Kolman, R. Busby & S.C. Ross, “Discrete Mathematical Structures”, 6th Edition, Pearson Education.
2. S. Lipschutz & M. Lipson, “Discrete Mathematics”, 3rd Edition, TMH.
3. J.P. Tremblay & R. S. Manohar, “Discrete Mathematical Structures with Applications to Computer Science, TMH, New York 1997.
4. Richard Hammack, “Book of Proof”, 2nd Edition, VCU Mathematics Text Book Series

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc18_cs53/preview

2. Link to topics related to course:

i. https://www.youtube.com/watch?v=xlUFkMKSB3Y

ii. https://www.youtube.com/watch?v=RMLR2JHHeWo

iii. https://www.youtube.com/watch?v=9AUCdsmBGmA

iv. https://www.youtube.com/watch?v=7cTWea9YAJE

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Discrete Computational Mathematics)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	2	1	2	2	2	1	2	3	3	2	**2.2**
CO-2	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
CO-3	3	3	2	1	1	3	3	3	3	3	1	1	**2.3**
CO-4	3	2	2	1	2	2	2	2	1	2	1	2	**1.8**
CO-5	3	2	2	1	2	3	3	2	1	3	2	1	**2.1**
CO-6	3	2	3	1	1	3	2	1	1	3	2	1	**1.9**
CO-7	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
Average	**3**	**2.6**	**2.3**	**1.1**	**1.6**	**2.6**	**2.6**	**1.7**	**1.9**	**2.6**	**2**	**1.4**	

Modeling and Simulation Techniques

COURSE CODE: 18B11CI413

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 2-0-0

Pre-requisite: Discrete Mathematics, Algorithm, Software Engineering

Course Objectives:

1. Student will model real-world systems and implement the model as a computer program
2. Student will learn model design and development comparison to analytical models.
3. Student will learn important methods of computing and statistics.
4. Student will learn important techniques of real world project development and management.
5. Student will learn to evaluate the performance of real-world systems by analyzing the output of the model under various conditions..

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	To learn the basic concepts, applications and terminology of computer simulation and modeling.	Familiarity
CO-2	To learn statistical methods of estimation and testing and other relevant concepts	Technical
CO-3	To explain the working and applications of different types of simulation such as Monte Carlo, VS. Discrete Event	Technical
CO-1	You will learn how to model a system and the execution of simulation tools.	Technical
CO-2	You will learn to analyze input data, its parameters, and the use of random number in a typical simulation study.	Technical
CO-3	Student will learn different techniques for the Verification and Validation of a simulation study	Technical

Course Contents:

Unit	**Contents**	**Lectures required**
1	**Introduction to Simulation and Modeling:** Simulation: Definition, Methods, Systems, Variability, Complexity, Advantages. Modelling: Definition, characteristics, description, categories.	**5**
2	**Statistical Concepts:** Hypothesis, Estimation, Statistical Significance, Error/Risks. Statistical tests, Bounds and Correlation. Input Data Modelling, Output Data Analysis.	**5**
3	**Discrete-Event Simulation, Monte Carlo Simulation:** Queuing System Model Components, Simulation Methodology, DES Example, Implementation, Arena Simulation. The Monte Carlo Method, Sensitivity Analysis	**8**
4	**Systems Modeling:** System Model Types, Modeling Methodologies and Tools, Analysis of Modeling and Simulation, Operation Research Methods, Coding the Model, Use of Pseudo Random Number Streams.	**8**
5	**Data Collection and Analysis :** Obtaining Data, Data Format, Representing Unpredictable Variability, Distributions, Bootstrapping,	**8**

	Fitting Statistical Distributions to Empirical Data	
6	**Verification, Validation, and Accreditation:** Definition and concepts, Difficulties, Confidence as Validity. Conceptual Model Validation, Data Validation, White-Box Validation, Black-Box Validation, Experimentation Validation, Solution Validation, Independent Verification and Validation	**8**
Total lectures		**42**

Suggested Text Book(s):

1. Modeling and Simulation: Exploring Dynamic System Behavior, Authors: Birta, Louis G., Arbez, Gilbert
2. Simulation (5th Edition) , Authors: Sheldon Ross.

Suggested Reference Book(s):

1. MODELING AND SIMULATION FUNDAMENTALS Theoretical Underpinnings and Practical Domains by John A. Sokolowski Catherine M. Banks.
2. Science in the Age of Computer Simulation by ERIC WINSBERG
3. Modeling and Simulation: The Computer Science of Illusion. By Raczynski, S..

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106104019/

2. Link to topics related to course:

i. https://nptel.ac.in/courses/106104019/1
ii. https://nptel.ac.in/courses/106104019/4
iii. https://nptel.ac.in/courses/106104019/26
iv. https://nptel.ac.in/courses/106104019/2Ev

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Data Simulation and Modeling Techniques)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	1	2	1	3	3	2	3	3	3	**2.5**

CO-2	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO-3	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO-4	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO-5	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO-6	3	3	3	1	2	3	3	3	3	3	3	3	**2.8**
Average	**3**	**3**	**3**	**2.3**	**2**	**2**	**3**	**3**	**2.2**	**3**	**3**	**3**	

Operating System

COURSE CODE: 18B11CI411

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: C/C++

Course Objectives:

1. To learn the mechanisms of OS to handle processes and threads and their communication.
2. To learn the mechanisms involved in memory management in contemporary OS.
3. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
4. To know the components and management aspects of concurrency management.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	An appreciation of the role of an operating system.	Familiarity
CO-2	Create processes and threads.	Assessment
CO-3	Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.	Assessment
CO-4	For a given specification of memory organization develop the techniques for optimally allocating memory to processes by Increasing memory utilization and for improving the access time	Assessment
CO-5	Design and implement file management system.	Assessment
CO-6	For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers. .	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.	**4**
2	**Process:** Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. **Thread:** Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.	**4**
3	**Process Scheduling:** Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, Priority, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF	**4**
4	**Inter-process Communication:** Critical Section, Race Conditions, Mutual	**10**

	Exclusion, Hardware Solution, Strict Alternation, Peterson‟s Solution, Lamport‟s Bakery Algorithm, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader‟s & Writer Problem, Dinning Philosopher Problem etc.	
5	**Deadlocks:** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker‟s algorithm, Deadlock detection and Recovery.	**4**
6	**Memory Management:** Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.	**6**
7	**Virtual Memory**: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).	**5**
8	**I/O Hardware:** I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms. **File Management:** Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. **Disk Management:** Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.	**5**
Total lectures		**42**

Suggested Text Book(s):

1. “Operating System Concepts” 9th Edition by Silberschatz, Peter Galvin, Greg Gagne, Wiley Asia Student Edition.
2. “Operating Systems: Internals and Design Principles” 9th Edition, William Stallings, Pearson.

Suggested Reference Book(s):

1. “Operating System: A Design-oriented Approach, 1st Edition by Charles Crowley, Irwin Publishing
2. “Operating Systems: A Modern Perspective” 2nd Edition by Gary J. Nutt, Addison-Wesley
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India.
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106108101//

2. Link to topics related to course:

   i. https://nptel.ac.in/courses/106106144/

   ii. https://www.class-central.com/course/udacity-introduction-to-operating-systems-3419

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Operating System)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Design and Analysis of Algorithms

COURSE CODE: 18B11CI412

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisite: Data structure and algorithms

Course Objectives:

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Apply important algorithmic design paradigms and methods of analysis.
5. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.	Technical
CO-2	Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.	Technical
CO-3	Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.	Technical
CO-4	Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. For a given problems of dynamic-programming and develop the dynamic programming algorithms, and analyze it to determine its computational complexity.	Technical
CO-5	For a given model engineering problem model it using graph and write the corresponding algorithm to solve the problems.	Technical
CO-6	Explain the ways to analyze randomized algorithms (expected running time, probability of error).	Technical
CO-7	Explain what an approximation algorithm is. Compute the approximation factor of an approximation algorithm (PTAS and FPTAS).	Technical

Course Contents:

Unit	Contents	Lectures required
1	Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior; Performance measurements of Algorithm, Time and space trade-offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters‟ theorem. Examples: Binary counter, Recursive Fibonacci in LogN	**10**
2	Fundamental Algorithmic Strategies: Brute-Force, Greedy, Dynamic Programming, Branch and Bound and Backtracking methodologies for the design	**8**

	of Algorithms; Illustrations of these techniques for Problem- Solving , Bin Packing, Knap Sack TSP. Heuristics characteristics and their application domains. Examples: Greedy and dynamic scheduling, Hoffman encoding, Dynamic programming: Longest common subsequence, Matrix chain multiplication, Dynamic programming: Bin packing, Knapsack, Dynamic programming: TSP, Branch and Bound TSP, Backtracking: SAT, Maze Sudoku solver, 8 queen	
3	Graph and Tree Algorithms: Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm. Examples: Best first search, Binomial heap MST: Prims, Kruskal, Fibonacci heap MST: Tarjan, Lazy decrease key implementation and Dijkstra, MaxFlow/ MinCut Ford-Fulkersion	**8**
4	Tractable and Intractable Problems: Computability of Algorithms, Computability classes – P, NP, NP-complete and NP-hard. Cook"s theorem, Standard NP-complete problems and Reduction techniques. Examples: Set Cover, Vertex cover, Map coloring, chromatic number	**8**
5	Advanced Topics: Approximation algorithms, Randomized algorithms, Class of problems, beyond NP – P SPACE, Examples: Approximate matrix inversion, Randomized Eigen vector computation	**8**
Total lectures		**42**

Suggested Text Book(s):

1. Rntroduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.

Suggested Reference Book(s):

1. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson.
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Second Edition,
3. Michael T Goodrich and Roberto Tamassia, Wiley.
4. Algorithms -- A Creative Approach, 3RD Edition, UdiManber, Addison-Wesley, Reading, MA.

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc18_cs20/preview
2. Link to topics related to course:
   i. https://nptel.ac.in/courses/106101060/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Design and Analysis of Algorithms)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	3	3	3	3	1	1	3	3	3	1	3	**2.5**
CO-2	2	2	2	2	2	1	1	1	2		1	2	**1.6**
CO-3	3	3	3	3	3	1	1	1	3	1	1	3	**2.2**
CO-4	2	2	2	2	2	2	1	1	2		2	2	**1.8**
CO-5	2	2	2	2	2	3	1	1	2	2	2	2	**1.9**
CO-6	2	2	2	2	2	2	1	2	2	2	2	2	**1.9**
CO-7	2	2	2	2	2	2	1	2	2	2	2	2	**1.9**
Average	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**1.7**	**1**	**1.6**	**2.3**	**2**	**1.6**	**2.3**	

Data Simulation Lab

COURSE CODE: 18B17CI473

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisite: None

Course Objectives:

1. Student will learn simulation of real world problems using python, numpy, scipy and simpy.
2. Students will learn structural development of complex system in terms of process, resources and lavels
3. Student will learn to use random number generator.
4. Students will learn to monitor and tally simulation results.
5. Students will apply simulation and modeling techniques in many real examples and develop projects.

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	Student will learn simulation of real world problems using python, scipy and simpy.	Familiarity
CO-2	Students will learn structural development of complex system in terms of process, resources and levels	Assessment
CO-3	Student will learn to use random number generator.	Assessment
CO-4	Students will learn to monitor and tally simulation re0sults.	Assessment
CO-5	Students will apply simulation and modelling techniques in many real examples.	Usage
CO-6	Students will apply simulation and modelling techniques in a real life project.	Usage

List of Experiments

S.No	**Description**	**Hours**
1	Simpy Getting started	**2**
2	Simple simulation	**2**
3	Simulation with tracing	**2**
4	Executing simulations event-by-event	**2**
5	Synchronizing simulation time with wallclock time	**2**
6	Event stepping of models with a GUI	**2**
7	Debugging	**2**
8	GUI for SimPy simulations	**2**
9	Plot for simulation	**2**
10	Statistical parameters	**2**
11	Random number	**2**
12	Inference and Output monitoring	**2**
13	MID SEM TEST	**2**
14	DES simulation on simpy	**2**
15	Simpy more features	**2**
16	Simpy internals	**2**
17	Example: MACHINE BREAKDOWN	**2**
18	Example: CAR WASH	**2**
19	Example: CELL PHONE NETWORK	**2**
20	Example: Resource pool	**2**

21	Group Project	**2**
22	Group Project	**2**
23	Group Project	**2**
24	Group Project	**2**
25	Group Project	**2**
26	Group Project	**2**
27	Group Project	**2**
28	End Semester Test	**2**
Total Lab hours		**56**

Suggested Books/Resources:

i. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3rd Edition, Prentice-Hall of India, 2010.
ii. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.
iii. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2nd Edition, Universities press, 2007
iv. https://nptel.ac.in/courses/106104019/26
v. https://nptel.ac.in/courses/106104019/2Ev

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	Average
CO1	3	3	3	1	2	1	3	3	2	3	3	3	**2.5**
CO2	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO3	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO4	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
CO5	3	3	3	3	2	2	3	3	2	3	3	3	**2.8**
C06	3	3	3	1	2	3	3	3	3	3	3	3	**2.8**
Average	**3**	**3**	**3**	**2.3**	**2**	**2**	**3**	**3**	**2.2**	**3**	**3**	**3**	

Operating Systems Lab

COURSE CODE: 18B17CI471

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisite: None

Course Objectives:

1. Be able to create sockets and analyze different (client/server) models.
2. Be able to create processes, threads, semaphores.
3. Be able to analyze different protocols.
4. Be able to learn how resources are being managed in Operating system.
5. Be able to manage system memory.

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO1	Understand basics of MATLAB syntax, functions and programming.	Familiarity
CO2	Generate and characterize various continuous and discrete time signals.	Familiarity
CO3	Perform the basic operations on the signals.	Computational skills
CO4	Design and analyze linear time-invariant (LTI) systems and compute its response.	Technical skills
CO5	Analyze the spectral characteristics of signals using Fourier analysis.	Technical skills
CO6	Analyze the systems using Laplace transform and Z-transform.	Technical skills

List of Experiments

S.No	**Description**	**Hours**
1	Process Handling	**2**
2	Zombie and Orphan Process	**2**
3	FCFS Scheduling Algorithm	**2**
4	SJF Scheduling Algorithm	**2**
5	Priority Scheduling Algorithm	**2**
6	Round-Robin Scheduling Algorithm	**2**
7	Process Groups	**2**
8	Inter-Process Communication	**2**
9	Shared Memory Concept	**2**
10	Peterson‟s Critical Section Problem Solution	**2**
11	Mutex	**2**
12	Semaphores	**2**
13	MID SEM TEST	**2**

Approved in Academic Council held on 28 July 2021

14	Safety Algorithm	**2**
15	Banker‟s Algorithm	**2**
16	Page Replacement algorithm: FIFO	**2**
17	Page Replacement algorithm: LRU	**2**
18	Page Replacement algorithm: Optimal Replacement	**2**
19	Disk Scheduling : C-SCAN	**2**
20	Disk Scheduling : C-LOOK	**2**
21	Group Project	**2**
22	Group Project	**2**
23	Group Project	**2**
24	Group Project	**2**
25	Group Project	**2**
26	Group Project	**2**
27	Group Project	**2**
28	END SEM TEST	**2**
Total Lab hours		**56**

Suggested Books/Resources:

1. Silberschatz and Galvin, Operating System Concepts.
2. Tanenbaum ,S.A Woodhull, Operating System :Design and Implementation
3. W. Stallings, Operating System:Internals and Design Principles

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Design and Analysis of Algorithms Lab

COURSE CODE: 18B17CI472

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisite: None

Course Objectives:

1. Student will understand the running time using time library functions. Learn to prepare table for input size vs. running time. Learn to measure best run and worst run of the experiments.
2. Students will learn to implement various types of design for an algorithms and compare the approaches.
3. Students will learn to implement network algorithms and their applications.
4. Students will learn to implement approximate algorithms for real world problems.
5. Students will learn to implement randomized solution for difficult real world problems.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO1	Student will understand the running time using time library functions. Learn to prepare table for input size vs. running time. Learn to measure best run and worst run of the experiments.	Technical
CO2	Students will learn to implement various types of design for algorithms and compare the approaches.	Technical
CO3	Students will learn to implement network algorithms and their applications.	Technical
CO4	Student will learn to implement classical NP problems	Technical
CO5	Students will learn to implement approximate algorithms for real world problems.	Technical
CO6	Students will learn to implement randomized solution for difficult real world problems.	Technical

List of Experiments

S.No	Description	Hours
1	Getting acquainted with time.h, clocktick, cputime, I/O time	2
2	Getting acquainted with worst case time	2
3	Getting acquainted with Average case time	2
4	Getting acquainted with recursive program	2
5	Recursive Fibonacci in log n	2
6	Greedy and dynamic scheduling	2
7	Hoffman encoding	2
8	Dynamic programming: Longest common subsequence, Matrix chain multiplication	2
9	Dynamic programming: Bin packing, Knapsack	2
10	Dynamic programming: TSP	2
11	Branch and Bound TSP	2
12	Backtracking: SAT, Maze	2

Approved in Academic Council held on 28 July 2021

13	Sudoku solver, 8 queen	2
14	MID sem TEST	2
15	Best first search	2
16	Binomial heap MST: Prims, Kruskal	2
17	Fibonacci heap MST: Tarjan	2
18	Lazy decrease key implementation and Dijkstra	2
19	MaxFlow/ MinCut Ford-Fulkersion	2
20	Set Cover, Vertex cover	2
21	Map coloring, chromatic number	2
22	Approximate matrix inversion	2
23	Randomized Eigen vector computation	2
24	Minor Project	2
25	Minor Project	2
26	Minor Project	2
27	Minor Project	2
28	Final test	2
Total Lab hours		**56**

Suggested Books/Resources:

1. Data Structures and Algorithms with Python, Lee and Hubbard.
2. Algorithm Design, 1ST Edition, Jon Kleinberg and ÉvaTardos, Pearson
3. ink to topics related to course:
    a. Python
    b. SciPy
    c. NumPy

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	3	3	3	1	1	3	3	3	1	3	**2.5**
CO2	2	2	2	2	2	1	1	1	2		1	2	**1.6**
CO3	3	3	3	3	3	1	1	1	3	1	1	3	**2.2**
CO4	2	2	2	2	2	2	1	1	2		2	2	**1.8**
CO5	2	2	2	2	2	3	1	1	2	2	2	2	**1.9**
CO6	2	2	2	2	2	2	1	2	2	2	2	2	**1.9**
Average	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**1.7**	**1**	**1.5**	**2.3**	**2**	**1.5**	**2.3**	

Web Technology Lab

COURSE CODE: 18B17CI474

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

L-T-P: 0-0-4

Pre-requisite: None

Course Objectives:

1. To develop the ability to design and implement web enabled applications.
2. The student shall acquire the skill to design and develop web based applications with high usability, scalability and efficiency.
3. They shall be exposed to various technologies required to design web sites
4. They shall acquire the skill to choose the technology to use based on the requirements and functionality of the web site

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO1	Basic PHP Concepts, PHP Operators, PHP Function, PHP Variables and Super globals.	Familiarity
CO2	Conditional Statements, Looping Statements, Array, Cookies, PHP Form, PHP Session, File Upload, File Handling, User login and Registration.	Usage
CO3	Database Connectivity, MySQL, MySQL connect, create DB/Table, Instructions such as select, where, order By, update and delete etc., encryption methods.	Familiarity
CO4	Create and save an XML document at the server, which contains 10users information. Write a program which takes User Id as input and returns the user details by taking the user information from the XML document.	Assessment
CO5	To get familiar with JavaScript, working with operators, Conditional Statements, looping statements, Alert Box, Confirm Box and Prompt Box, Functions, Array, event handler, regular expressions and modifiers, Cookie and form validations.	Assessment
CO6	Validate the registration, user login, user profile and payment by credit card pages using JavaScript.	Usage

List of Experiments

S.No	Description	Hours
1	Basic PHP Concepts, PHP Operators, PHP Function, PHP Variables and Super globals,	6
2	Conditional Statements, Looping Statements, Array, Cookies, PHP Form, PHP Session, File Upload, File Handling, User login and Registration	6
3	Database Connectivity, MySQL, MySQL connect, create DB/Table, Instructions such as select, where, order By, update and delete etc., encryption methods.	6
4	Project-To develops and implement, and demonstrate Database Driven Websites through a project that meet stated specifications.	4
5	Create and save an XML document at the server, which contains 10users information. Write a program which takes User Id as input and returns the user details by taking the user information from the XML document.	6

6	To get familiar with JavaScript, working with operators, Conditional Statements, looping statements, Alert Box, Confirm Box and Prompt Box, Functions, Array, event handler, regular expressions and modifiers, Cookie and form validations.	6
7	Validate the registration, user login, user profile and payment by credit card pages using JavaScript.	4
8	Bean Assignments a. Create a JavaBeans which gives the exchange value of INR (Indian Rupees) into equivalent American/Canadian/Australian Dollar value. b. Create a simple Bean with a label which is the count of number of clicks. Then create a Bean info class such that only the count properly is visible in the property Window. c. Create two Beans – a) Keypad b) Display pad. After that integrate the two Beans to make it work as a calculator. d. Create two Beans Traffic Light (implemented as a label with only three background colors-red, green, yellow) and Automobile (Implemented as a Text Box which states its state/movement). The state of the Automobile should depend on the following Light Transition table	8
9	XML Concepts, XML Elements and Attributes, DTD and Schema, XML with CSS.	4
10	Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using servlets and cookies. Hint: User"s information (user id, password, credit card number) would be stored in web. Xml. Each user should have a separate shopping cart	6
Total Lab hours		**56**

Suggested Books/Resources:

1. Web Enabled commercial Application development using HTML,DHTML, Java Script, Perl CGI" by Ivan Bayross, BPB Publication
2. "Internet and World Wide Web – How to Program" by Deitel, Deitel and Nieto ,Pearson Education Asia Publication
3. "PHP and MYSQL Manual" by Simon Stobart and Mike Vassileiou
4. "PHP and MYSQL Web Development" by Luke Welling and Laura Thomson(Pearson Education

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	3	3	2	2	3	2	2	2	3	1	3	2.4
CO2	3	3	3	2	3	2	3	2	2	3	1	3	2.5
CO3	3	3	3	2	2	3	1	2	3	3	1	3	2.4
CO4	3	3	3	2	3	3	2	2	3	3	1	3	2.6
CO5	3	3	3	2	3	3	2	2	3	3	1	3	2.6

CO6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Computer Graphics

COURSE CODE: 18B11CI515

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

L-T-P: 3-0-0

Pre-requisites: Elementary Knowledge about the algorithms, programming, matrices and linear algebra is required

Course Objectives:

1. To learn and understand the basics of computer graphics applications and graphics devices
2. To learn and understand the geometric figure drawing algorithm on graphic device
3. To learn and understand the Two-Dimensional transformations
4. To learn and understand the Three-Dimensional transformations
5. To understand the concepts of solid modelling and representation
6. To learn about the Visible-Surface, Illumination and Shading

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO 1	Student will learn about the overview of computer graphic applications and graphics devices (Display Technologies, Raster Refresh (Raster- Scan), CRT, LCD displays, etc.)	Familiarity
CO 2	Student will learn about the scan conversion - lines, circles and Ellipses, filling, clipping and aliasing	Usage
CO 3	Student will learn about the Two-Dimensional transformations and matrix representation of 2D Transformations (Translations, Rotation, Reflection, Scaling and Combined Transformation) and Window-to- Viewport transformations	Familiarity
CO 4	Student will learn about the Three-Dimensional transformations and viewing in 3D	Assessment
CO 5	Student will learn about the solid modelling: representing solids, regularized Boolean Set operations, primitive instancing, sweep representations, spatial-partitioning representations - Octree representation, B-Reps and Constructive Solid Geometry	Assessment
CO 6	Student will learn about the visible surface detection, illumination and shading	Usage

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Computer Graphics: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Colour CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays, Touch screen, Graphics Primitives.	4
2	Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms, Scan Converting Lines, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons, edge data structure, Clipping Lines algorithms– Cyrus-Beck,	14

	Cohen- Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components	
3	Two-Dimensional Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window- to-Viewport Transformations	**6**
4	Three-Dimensional Transformations and Viewing in 3D: Introduction, Three-Dimensional Scaling, Three-Dimensional Shearing, Three- Dimensional Rotation, Three-Dimensional Reflection, Three- Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.	**10**
5	Solid Modelling: Representing Solids, Regularized Boolean Set Operations, Primitive Instancing, Sweep Representations, Spatial- Partitioning Representations: Octree representation, B-Reps, Constructive Solid Geometry, Comparison of Representations	**3**
6	Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter"s algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing,	**3**
	comparison of the methods	
7	Illumination and Shading: Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Atmospheric attenuation, Phong"s model, Gouraud shading, some examples.	**2**
8	Image Manipulation and Storage: What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.	**Self- Learning**
Total Lectures		**42**

Suggested Text Book(s):

1. Hearn Donald & Baker, M. Pauline (1990). "Computer Graphics – C Version" Prentice Hall of India Pvt. Ltd.

Suggested Reference Book(s):

1. Foley, J.D. & Dam, A. Van (1982), "Fundamentals of Interective Computer Graphics", Addison-Wesley
2. Harrington, S. (1983). "Computer Graphics: A Programming Approach"" Mc-Graw Hill Book Co.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Sr. No.	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	2	2	2	2	1	2	1	2	1	2	2	**1.8**
CO-2	3	3	3	3	3	1	2	1	3	3	3	3	**2.6**
CO-3	3	3	3	3	3	1	1	1	3	1	3	3	**2.3**
CO-4	3	3	3	3	3	3	1	1	3	1	3	3	**2.5**
CO-5	2	2	2	2	2	2	2	1	2	2	2	2	**1.9**
CO-6	2	2	2	2	2	2	2	1	2	2	2	2	**1.9**
Average	**2.5**	**2.5**	**2.5**	**2.5**	**2.5**	**1.7**	**1.7**	**1**	**2.5**	**1.7**	**2.5**	**2.5**	

Formal Languages and Automata

COURSE CODE: 18B11CI513

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

: 3-0-0

Pre-requisites: Graduate level courses in C programming, algorithms and complexity of algorithms are desirable.

Course Objectives:

1. To introduce the students to the mathematical foundations of computation including automata theory
2. To introduce the students about the theory of formal languages and grammars
3. To introduce the students about the theory of Automata which includes Finite Automata, Push down Automata and Turing Machines
4. To introduce the students about the notions of algorithm, decidability
5. To introduce the students about the Complexity and computability

Course Outcome (CO):

S.No.	Course Outcomes	Level of Attainment
CO-1	Broaden knowledge of the fundamental mathematical and computational principles that are the foundation of computer science	Familiarity
CO-2	Understand the concept of Deterministic Finite Automata and Non-Deterministic Finite Automata	Usage
CO-3	Understand how to minimize the states, usage Moore and Mealy Machine	Assessment
CO-4	Understand how to use the context free grammars in languages and how to derive parse trees and solve ambiguity problems	Assessment
CO-5	Understand Normal forms for Context Free Grammar"s Chomsky and Greibach Normal Forms	Assessment
CO-6	Understand the Push Down Automaton algorithm	Assessment
CO-7	Understand how the push down automata will accept arbitrary context free languages. To understand the properties of CFG To understand the determinism and parsing. To understand different parsing methodologies	Assessment
CO-8	Understand the basic concepts of Turing Machine, configuration of Turing Machine, computing with the Turing Machine	Usage
CO-9	Understand multiple tapes, two way infinite tape concepts, the real computers random access memories working, concept of non-deterministic Turing machines	Assessment
CO-10	Understand the computational power of languages, numerical functions applied to Turing machines, numerical functions applied to Turing machines, various mathematical models applied to Turing machines, the concept of halting problem, undecidable problems about Turing machines and grammars, properties of recursive languages, concept of polynomial decidable	Assessment

Course Contents:

Unit	Contents	Lectures required
1	Introduction &Motivation, Infinite Sets, Closures, Alphabets, Languages & Representation	**3**
2	Deterministic finite automata, Non-Deterministic finite automata, Closure Properties & Equivalences, State Minimization, Moore and Mealy Machine	**6**
3	Formal Languages, Regularity, Regular Grammars, Regular Expressions and Finite Automata	**6**
4	Context Free Grammars, Parse Trees & Ambiguity, Chomsky and Greibach Normal forms	**8**
5	Push Down Automata, Equivalence of PDA and CFG, Properties of context free languages, Determinism & Parsing DCFG, Top-down & Bottom-up Parsing	**6**
6	Turing Machine-Introduction, Notations, Recursive and Recursively Enumerable Language, Extensions of Turing machines, Non-deterministic Turing machines, Primitive Recursive Functions, Mu-recursive functions	**8**
7	Church-Turing Thesis & Universal Turing machines, Halting problem, Undecidable problems, Properties of Recursive languages	**2**
8	The Complexity Class P, Satisfiability, The Complexity Class NP, NP Completeness and Reducibility NP complete problems, Cook's Theorem, NP Complete Problems	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Elements of the Theory of Computation. Harry Lewis, Christos Papadimitriou, Second Edition, Pearson Education, 1998

2. Theory of computer Science: Automata, Language and Computation - KLP Mishra N Chandra Sekhran - PHI, 3rd edn.

Suggested Reference Book(s):

1. Formal Languages and Automata- Peter Linz , Narosa Pub. 4th edn.

2. M. Sipser, Introduction to the Theory of Computation, Thomson Asia, 1997.

3. J. E. Hopcroft, R. Motwani and J. D. Ullman, Introduction to Automata Theory, Languages and Computation, Pearson, 2001.

4. D. C. Kozen, Automata and Computability, Springer-Verlag, 1997.

Other useful resource(s):

1. https://nptel.ac.in/courses/111103016/

2. Link to topics related to course:

- **i.** https://ocw.mit.edu/courses/mathematics/18-404j-theory-of-computation-fall-2006/
- **ii.** http://www.aduni.org/courses/theory/
- **iii.** http://www.cse.iitd.ernet.in/~sak/courses/toc/
- **iv.** http://cse.iitkgp.ac.in/~abhij/course/theory/FLAT/Spring13/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes (Formal Languages and Automata)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	2	2	2	2	3	3	2	1	3	2	1	**2.2**
CO-2	3	3	3	2	2	3	3	3	1	3	3	1	**2.5**
CO-3	3	3	3	2	2	3	2	3	2	2	3	2	**2.5**
CO-4	3	3	3	2	2	2	3	2	2	3	2	2	**2.4**
CO-5	3	3	3	2	2	2	2	2	1	2	2	1	**2.1**
CO-6	3	3	3	3	2	3	3	2	2	3	2	2	**2.6**
CO-7	3	3	3	3	2	3	3	2	2	3	2	2	**2.6**
CO-8	3	3	3	2	2	3	3	3	2	3	3	2	**2.7**
CO-9	3	3	3	2	2	3	3	3	2	3	3	2	**2.7**
CO-10	3	3	3	2	2	3	3	3	2	3	3	2	**2.7**
Average	**3**	**2.9**	**2.9**	**2.2**	**2**	**2.8**	**2.8**	**2.5**	**1.7**	**2.8**	**2.5**	**1.7**	

Computer Organization and Architecture

COURSE CODE: 18B11CI514
COURSE CREDIT: 3
CORE/ELECTIVE: CORE
L-T-P: 3-0-0

Pre-requisites: Basic Understanding of Computer System

Course Objectives:

1. To calculate the performance of a modern digital computer from parameters such as processor speed, cycles per instruction.
2. To understand the fixed-point and floating-point numbers are represented in a computer.
3. Discuss about pipelining in a processor functions and describe how hazards are resolved in various ways.
4. Wide understanding of memory organization and management in a modern digital computer, including virtual and physical memory, address translation, multilevel, unified, and multi-way set-associative caches, the translation-look-aside buffer (TLB), and the page table.
5. To understand the working strategies of parallel processing and multi-core computers.

Course Outcomes:

S. No	Course Outcome (Computer Organization & Architechture)	Level of Attainment
CO-1	To learn the basic concepts, terminology and evolution in computer organization and architecture	Familiarity
CO-2	Understanding the computer architecture and computer arithmetic.	Assessment
CO-3	Understanding of the computer memory and the issues related to memory.	Assessment
CO-4	Understanding the concept of memory I/O, interrupt handling and DMA.	Assessment
CO-5	Learn the organization of Processor and the concept of pipelining.	Assessment
CO-6	Learning concepts of Parallel processing and related issues.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Basic Structure of Computers: Functional units:** Basic operational concepts – Bus structures – Performance and metrics – Instructions and instruction sequencing – Hardware – Software Interface – Instruction set architecture – Addressing modes – RISC – CISC.	7
2	**Basic Processing Unit of Computers:** ALU design – Integer Representation, floating point and its Arithmetic. Interconnection Structure, Bus interconnection Structure, point to point interconnection PCI express structure.	7
3	**Computer Memory:** Cache Memory: Overview of computer Memory, Cache memory principles and its elements to design P-4 Cache organization.	6
4	**Internal and External Memory:** Semiconductor Memory: Semiconductor Main Memory, Error correction. External Memory: Magnetic Disk and Tape, RAID, SSD, Optical Memory.	
5	**Input/output:** Access of I/O devices, I/O ports, I/O control mechanisms - Program controlled I/O Interrupt controlled I/O and DMA controlled I/O I/O interfaces Program controlled I/O, Interrupt controlled I/O, and DMA controlled I/O, I/O interfaces - Serial port,	6

	Parallel port, PCI bus, SCSI bus, USB bus	
6	**Central Processing Unit:** Processor Structure and Functions: Processor structure, register organization, instruction cycle and pipelining.	4
7	**Instruction level Parallelism:** Overviews, design issues, vector processing	3
8	**Parallel Processing:** Multiple Processor Organization, Symmetric Multi-processors, cache coherence, multithreading, clusters, non- uniform memory access, and vector computation.	5
Total lectures		**42**

Suggested Text Book(s):

1. William Stallings, Computer Organization & Architecture - Designing for Performance Eighth Edition, Pearson, 2010. ISBN 978-81-317-3245-8.

2. M. Morris Mano, Computer System Architecture, Pearson.

Suggested Reference Book(s):

1. John L. Hennessy and David A. Patterson ,Computer Architecture: A Quantitative Approach, Fourth Edition, Morgan Kaufmann Publishers

2. M. Morris Mano , Computer System Architecture, Third Edition, Pearson Education Inc

3. Luiz Andre Barroso and Urs Holzle;The Datacenter as a Computer – An Introduction to the Design of Warehouse Scale Machines Morgan and Claypool Publishers

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/syllabus/106103068/
2. Link to topics related to course: https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs):

Course Outcomes (Computer Organization and Architecture)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	2	2	2	2	2	2	1	2	1	1	2	**1.8**
CO-2	3	3	3	3	1	3	1	2	2	2	2	2	**2.3**
CO-3	3	2	3	3	2	2	2	2	2	2	2	2	**2.3**
CO-4	3	3	3	2	2	3	2	2	3	2	2	2	**2.4**
CO-5	3	2	2	3	2	3	3	2	3	2	2	2	**2.4**
CO-6	3	2	2	2	3	2	2	2	3	1	2	2	**2.2**
Average	**3**	**2.3**	**2.5**	**2.5**	**2**	**2.5**	**2**	**1.8**	**2.5**	**1.7**	**1.8**	**2**	

Computer Graphics Lab

COURSE CODE: 18B17CI575
COURSE CREDITS: 2
CORE/ELECTIVE: CORE
: 0-0-4

Pre-requisite: None

Course Objectives:

1. Using OpenGL for Graphics
2. Programming User-interface issues
3. Concepts of 2D & 3D object representation
4. Implementation of various scan & clipping algorithms
5. 2D modelling
6. Implementation of illumination model for rendering 3D objects
7. Visibility detection & 3D viewing
8. Implementation of a project based on learned concepts.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Using OpenGL for Graphics	Familiarity
CO-2	Programming User-Interface Issues	Assessment
CO-3	Concepts of 2D & 3D object representation	Assessment
CO-4	Implementation of various scan and clipping algorithms	Assessment
CO-5	2 D Modelling	Assessment
CO-6	Implementation of a project using learned models	Usage

List of Experiments:

S. No	Description	Hours
1	Basics of OpenGL: Draw a line, rectangle and pentagon	2
2	Implement DDA Line Algorithm	2
3	Implement Bresenham"s Line drawing Algorithm	2
4	Implement Bresenham"s Circle drawing Algorithm	2
5	Implement Bresenham"s Ellipse drawing Algorithm	2
6	Implement Boundary Fill Algorithm	2
7	Implement Flood Fill Algorithm	2
8	Implement Cohen Sutherland Line clipping algorithm	2
9	Implement Liang Barsky Line clipping algorithm	2
10	Implement Sutherland-Hodgman polygon clipping algorithm	2
11	Implement Nicholl-Lee-Nicholl Line clipping algorithm	2
12	Implement Weiler-Atherton polygon clipping algorithm	2
13	MID SEM TEST	2
14	Implement 2D Translation Transformation	2
15	Implement 2D Rotation Transformation	2
16	Implement 2D Scaling Transformation	2
17	Implement 2D Shear Transformation	2
18	Implement 3D Translation Transformation	2
19	Implement 3D Rotation Transformation	2

20	Implement 3D Scaling Transformation	2
21	Group Project	2
22	Group Project	2
23	Group Project	2
24	Group Project	2
25	Group Project	2
26	Group Project	2
27	Group Project	2
28	END SEM TEST	2
Total Lab Hours		**56**

Minor Project(s) – (*Only for 2 credit lab)*

1. Interactive 3D maze
2. Scene Recreation
3. Real-time rendering Technology

Suggested Books/Resources:

1. Donald Hearn and M. Pauline Baker. “Computer Graphics with OPENGL” 3rd Edition Pearson Publishers, 2011.
2. James D. Foley, Van Adams, K.Fenier and F. Hughes, “Computer Graphics-Principle and Practices”, 3rd Edition Pearson Publishers, 2002.
3. Harrington, S. “Computer Graphics: A Programming Approach” Mc-Graw Hill Book Co.

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Computer Organization and Architecture Lab

COURSE CODE**:** 18B17CI574
COURSE CREDITS: 1
CORE/ELECTIVE: CORE
: 0-0-2

Pre-requisite: None

Course Objectives:

1. To get familiar with the working scenario of logic gates.
2. Understanding the way in which arithmetic operations are done.
3. Structure of ALU and its Design
4. Understanding of Memory Design and its issues

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	Ability to understand basic structure of computer.	Assessment
CO-2	Ability to perform arithmetic operations on computer.	Assessment
CO-3	Ability to understand the memory concepts including Cache.	Assessment
CO-4	Familiarity with CPU design.	Familiarity

List of Experiments:

S. No	**Description**	**Lab Hours**
1	To Study And Verify The Truth Table Of Logic Gates	2
2	Realization Of A Boolean Function	2
3	Adders And Subtractors	2
4	Multiplexer And Demultiplexer	2
5	Registers and Counters	2
6	Booth"s Multiplier	2
7	Design Of ALU	4
8	Memory Design	4
9	Associative Cache Design	4
10	Direct Mapped Cache Design	2
11	CPU Design	2
Total Lab Hours		**28**

Suggested/Resources:

1. Computer Organization & Design: The Hardware/Software Interface - David Patterson and John Hennessey.
2. William Stallings, Computer Organization & Architecture - Designing for Performance Eighth Edition, Pearson, 2010. ISBN 978-81-317-3245-8.
3. Dr. M. Usha, T. S. Srikanth, “Computer System Architecture and Organization”,First Edition, Wiley-India.
4. “Computer Organization” by ISRD Group, Tata McGraw-Hill.
5. Link to topics related to course:
   i. http://cse10-iitkgp.virtual-labs.ac.in/index.html

Approved in Academic Council held on 28 July 2021

Course Outcomes (COs) contribution to the Programme Outcomes (POs):

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	Average
CO-1	3	2	2	2	2	3	2	2	2	2	2	2	**2.2**
CO-2	3	3	3	2	2	3	3	2	2	2	1	1	**2.3**
CO-3	3	3	3	2	2	2	3	2	2	2	3	2	**2.4**
CO-4	3	3	2	2	2	2	2	2	3	2	2	3	**2.3**
Average	**3**	**2.8**	**2.5**	**2**	**2**	**2.5**	**2.5**	**2**	**2.3**	**2**	**2**	**2**	

Multimedia Lab

COURSE CODE: 18B1WCI575
COURSE CREDITS: 1
CORE/ELECTIVE: CORE
: 0-0-2

Pre-requisite: None

Lab Course Objectives: This course focuses on the development of applications that manipulate media assets. Significant time is spent on intermediate to advanced programming and scripting as well as the synchronization of aural and graphical components. Students are requited to plan, design and implement lab projects.

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	To learn how to design and develop multimedia for real world e-learning.	Familiarity
CO-2	To learn how to implement multimedia e-leaning in Macromedia Flash.	Assessment
CO-3	Students will learn to create multi-user multimedia applications.	Assessment
CO-4	Students will learn to create a 3D multimedia application.	Assessment
CO-5	Students will learn to write interactive programs on the Android.	Assessment
CO-6	Students will learn to analyze and design's to interact with other apps on the device	Usage

List of Experiments:

S. No	**Description**	**Hours**
1	Write a program to justify a text entered by the user on both left and right hand side. for example the test " An architect may have a graphics program to draw an entire building but be interested in only ground floor", can be justified in 30 columns. An architect may have a graphics programs draw an entire building but interested in ground floor.	**2**
2	Study the notes of a piano and stimulate them using the keyboard and store them in file.	**1**
3	Write a program to read a paragraph and store it to a file name suggested by the author. Devise a routine to produce the animation effect of a square transforming to a triangle and then to a circle	**2**
4	To implement text and image compression algorithms: Huffman coding, Lempel-Ziv algorithm, Lempel–Ziv–Markov chain algorithm (LZMA), run length encoding, entropy encoding, area image compression, DPCM and predictive coding	**3**
5	Write a program to show a bitmap image on your computer screen.	**2**
6	Create a web page for a clothing company which contains all the details of that company and at least five links to other web pages.	**2**
7	Write a program by which we can split mpeg video into smaller pieces for the purpose of sending it over the web or by small capacity floppy diskettes and then joining them at the destination.	**4**
8	Write a program to simulate the game of pool table.	**4**
9	Write a program to simulate the game mine sweeper.	**4**
10	Write a program to play "wave" or "midi" format sound files.	**4**

Total Lab Hours	**28**

Suggested/Resources:

1. Flash MX Action Script Programming, Robert Reinhardt and Joey Lott, Wiley.
2. Flash 5 Magic with Action Script, J. Scott Hamlin and David J. Emberton, Techmedia
3. Beginning Android Application Development, Wei-Meng Lee, paperback
4. An introduction, Villamil & Molina, Multimedia Mc Milan, 1997
5. Multimedia: Sound & Video, Lozano, 1997, PHI, (Que)
6. Multimedia: Production, planning and delivery, Villamil & Molina,Que, 1997
7. Multimedia on the PC, Sinclair,BPB
8. Multimedia: Making it work, Tay Vaughan, fifth edition, 1994, TMH.
9. Multimedia in Action by James E Shuman, 1997, Wadsworth Publ.,
10. Multimedia in Practice by Jeff coate Judith, 1995,PHI.
11. Multimedia Systems by Koegel, AWL
12. Multimedia making it Work by Vaughar, etl.
13. Multimedia Systems by John .F. Koegel, 2001, Buford.
14. Multimedia Communications by Halsall & Fred, 2001,AW
15. Link to topics related to course:
    a. https://nptel.ac.in/courses/117105083/

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	2	1	1	2	1	2	2	1		2	1	1	**1.5**
CO2	1	1			1	1	1	1	1	1	1	1	**1**
CO3	1	1	1		1	1	1	1	1	1	1	1	**1**
CO4	1	1	1	1	1	1	1	1	1	1	1	1	**1**
CO5	1	1	1	1	1	1	1	1		1	1	1	**1**
CO6	1	1	1		1	1	1			1	1	1	**1**
Average	**1.2**	**1**	**1**	**1.3**	**1**	**1.2**	**1.2**	**1**	**1**	**1.2**	**1**	**1**	

DETAILED COURSE DESCRIPTIONS

ELECTIVE I

Data Compression

COURSE CODE: 18B1WCI532

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Digital image processing, UG mathematics, vectors, basic programming skills

Course Objectives:

1. To provide students with contemporary knowledge in Data Compression and Coding.
2. To equip students with skills to analyze and evaluate different Data Compression and Coding methods.
3. Analyze the operation of a range of commonly used Coding and Compression techniques
4. Identify the basic software and hardware tools used for data compression.
5. Identify what new trends and what new possibilities of data compression are available.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To understand the statistical basis for and performance metrics for lossless compression.	Familiarity
CO-2	To understand the conceptual basis for commonly used lossless compression techniques	Assessment
CO-3	To understand how to use and evaluate several readily available implementations of those techniques	Assessment
CO-4	To understand the structural basis for and performance metrics for commonly used lossy compression techniques	Assessment
CO-5	To understand the conceptual basis for commonly used lossy compression techniques	Assessment
CO-6	To implement graph theory in compression methodologies for images in MATLAB	Usage
CO-7	To understand image compression techniques‟ case studies	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	**Introduction**: Definitions, Historical background, Applications, Taxonomy	2
2	**Intuitive Compression:** Run-Length Encoding, RLE Text Compression, RLE Image Compression, Move-to Front Coding, Scalar Quantization	4
3	**Statistical Methods Information:** Theory Concepts, Variable- Size Codes, Prefix Codes, Golomb Codes, The Kraft-MacMillan Inequality, The Counting Argument, Shannon-Fano Coding, Huffman Coding, Adaptive Huffman Coding, MNP5, MNP7, Arithmetic coding, Adaptive Arithmetic Coding, QM Coder, Text Compression, Context-Tree Weighting	5
4	**Dictionary Methods:** String Compression, Simple Dictionary Compression, LZ77 (Sliding Window), LZSS, Repetition Times, QIC-122, LZX, File Differencing: VCDIFF, LZ78, LZFG, LZRW1,LZRW 4, LZW, LZMW, LZAP, LZY, LZP,	6

	Repetition Finder, UNIX Compression, The V.42bis Protocol, XML Compression: XMill, EXE Compressors, CRC, Data Compression Patents	
5	**Image Compression:** Approaches to Image Compression; Image Transforms, Orthogonal Transforms. The Discrete Cosine Transform JPEG, JPEG-LS. Progressive Image Compression, JBIG, JBIG2, Simple Images: EIDAC, Vector Quantization, Adaptive Vector Quantization, Block Matching, Block Truncation Coding, Context-Based Methods, FELICS, Progressive FELICS, Differential Lossless Compression	**6**
6	**Wavelet Methods:** Fourier Transform, The Frequency Domain, Fourier Image, Compression, Multiresolution Decomposition, The Laplacian Pyramid, SPIHT, CREW. EZW, DjVu, JPEG 2000	**6**
7	**Video Compression:** Analog Video , Composite and Components Video, Digital Video, Video Compression, MPEG, MPEG-4, H.261 **Audio Compression:** Sound, Digital Audio, The Human Auditory System , μ-Law and A-Law Companding, ADPCM Audio Compression, MLP Audio, Speech Compression, Shorten MPEG-1 Audio Layers	**7**
8	**Other Methods and Application:** Zip and Gzip, PNG, The Burrows-Wheeler Method, Symbol Ranking, ACB, SortBased Context Similarity, Sparse Strings, Word-Based Text Compression, Textual Image Compression, Dynamic Markov Coding, FHM Curve Compression, Sequitur, Triangle Mesh Compression: Unicode Compression	**6**
Total Lectures		**42**

Suggested Text Book(s):

1. David Salomon , A Concise Introduction to Data Compression, 1st edition, Springer, 2008
2. Sayood, Khalid, Introduction to Data Compression, 3rd Edition, Morgan Kaufmann, 2006

Suggested Reference Book(s):

1. David Salomon, G. Motta, D. Bryan, Data Compression: The Complete Reference, 4nd edition, Springer(2006)
2. D.C. Hankerson, Greg A. Harris , Peter D. Johnson Jr, Introduction to Information Theory and Data Compression, Second Edition, Chapman & Hall/CRC; 2 edition 2003
3. Anderson, J.B. and Mohan, S., Source and Channel Coding, Kluwer, 1991.
4. Gersho, A. and Gray, R.M., Vector Quantization and Signal Compression, Kluwer, 1992.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/108104098/41
2. Link to topics related to course:
   i. https://london.ac.uk/sites/default/files/study-guides/data-compression.pdf
   ii. http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE523.htm
   iii. http://www.nhu.edu.tw/~chun/CS-ch15-Data%20Compression.pdf
   iv. http://apachetechnology.in/ati/www/KC/dw/Saloman%20-%20Data_Compression_Complete_Reference.pdf

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2

3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

Course Outcomes (Data Compression)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	1	1	3	2	3	3	3	3	1	**2.4**
CO-2	2	2	2	2	3	3	2	3	3	3	1	3	**2.4**
CO-3	3	3	3	3	3	3	3	3	3	3	3	3	**3**
CO-4	3	3	2	3	3	3	2	1	1	3	3	3	**2.5**
CO-5	3	3	3	1	2	3	2	3	3	3	3	2	**2.6**
CO-6	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-7	3	3	3	3	3	3	3	3	3	3	3	3	**3**
Average	**2.7**	**2.7**	**2.6**	**2.3**	**2.6**	**3**	**2.3**	**2.7**	**2.7**	**3**	**2.4**	**2.6**	

Principles of Programming Languages

COURSE CODE: 18B1WCI533

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Familiarity with any one of the procedural or object oriented language such as C/C++/Java is mandatory.

Course Objectives:

1. Compare programming languages.
2. Describe the main principles of imperative, functional, object oriented and logic oriented programming languages;
3. Recite the high points of programming language history.
4. Read the central formalisms used in the description of programming languages.
5. Assess programming languages critically and in a scientific manner;
6. Analyze the principles of an imperative, functional, object oriented or logic oriented programming language; and
7. Use a formalism to describe a programming language.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn major programming paradigms and techniques involved in design and implementation of modern programming languages. To learn the structure of a compiler and interpretation. To learn syntax and symantic of programming language.	Familiarity
CO-2	To learn the structured and object oriented programming paradigm.	Familiarity
CO-3	To different programming paradigm to improving the clarity, quality, and development time of a program (structured programming). To learn Haskell (an advanced purely-functional programming style and lambda calculus (for variable binding and substitution).	Assessment
CO-4	To learn To understand basic logic programming through Prolog. Case study of a logic programming language – Prolog knapsack	Usage
CO-5	To learn the concurrency in programming languages, Exception handling and Scripting languages	Familiarity
CO-6	Case study of a markup language – XML. Common web development languages & technologies – XML, JavaScript, AJAX, Mashups, etc.	Assessment

Course Contents:

Unit	Contents	Lectures required
1	Introduction to the course Introduction to major programming languages History of major programming languages. Introduction to the programming language paradigms	4
2	Compilation and Interpretation Syntax of a programming language Programming language semantics	5

	Structured Programming Procedure Activations	
3	Type systems in programming languages Object oriented programming language features Java Virtual Machine	**5**
4	Introduction to functional programming paradigm Case study of a functional programming language – Haskell Introduction to the lambda calculus Introduction to logic programming paradigm Case study of a logic programming language – Prolog	**16**
5	Introduction to concurrency in programming languages Exception handling Scripting languages	**7**
6	Markup languages Case study of a markup language – XML Common web development languages & technologies XML, JavaScript, AJAX, Mashups, etc.	**4**
Total Lectures		**42**

Suggested Text Book(s):

1. Programming Languages: Concepts & Constructs, 2nd Edition by Ravi Sethi; Pearson Education Asia
2. Programming Language Principles and Paradigms 2nd Edition, Tucker, Allen B. Michael and Noonan Robert E. , TMH 2007

Suggested Reference Book(s):

1. Programming Languages and Paradigms, D. A. Watt., Prentice-Hall,1990
2. Essentials of Programming Languages, Daniel Friedman, Mitchell Wand, and Christopher Haynes, MIT Press (Indian edition - Prentice Hall, India)
3. Concepts of Programming Languages, Robert W. Sebesta, Pearson Education Asia
4. Programming Languages: Design & Implementation, Pratt & Zelkowitz, PHI Pub. (Latest Edition)
5. Programming Languages: Principles and Practices, Kenneth C. Louden, Thomson Press

Other useful resource(s):

1. https://nptel.ac.in/syllabus/106102067/
2. http://www.nptelvideos.in/2012/11/principles-of-programming-languages.html
3. https://nptel.ac.in/courses/106102067/
4. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-821-programming-languages-fall-2002/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes (Principles of Programming Languages)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	2	3	1	2	3	2	2	1	1	1	3	**2**

CO-2	3	2	3	1	1	3	2	3	2	1	1	3	**2.1**
CO-3	3	2	3	1	1	2	2	2	1	1	1	3	**1.8**
CO-4	3	3	3	1	1	2	2	2	1	1	1	3	**1.9**
CO-5	3	2	3	2	2	3	2	2	1	1	1	3	**2.1**
CO-6	3	2	3	2	2	3	2	2	2	1	1	3	**2.2**
Average	**3**	**2.2**	**3**	**1.3**	**1.5**	**2.7**	**2**	**2.2**	**1.3**	**1**	**1**	**3**	

Java Programming

COURSE CODE: 18B1WCI534

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Object-Oriented Systems and Programming

Course Objectives:

1. Using Graphics, Animations and Multithreading for designing Simulation and Game based applications.
2. Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling.
3. Design and develop Web applications
4. Designing distributed applications using Remote Method Invocation (RMI)

Course Outcomes:

S. No	Course Outcomes	Level of Attainment
CO-1	To learn the graphics and animation on the web pages, using Java Applets. To learn and design a full set of Event driven UI widgets and other components, including windows, menus, buttons, checkboxes, text fields, scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT) & Swings	Usage
CO-2	To learn Java Data Base Connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through Java programs.	Usage
CO-3	To learn the server side programming using Servlets and JSP.	Usage
CO-4	To learn the invocation of the remote methods in an application using RMI	Usage

Course Contents:

Unit	Contents	Lectures required
1	Applet Programming: Structure and Life Cycle of a Java Applet, Graphics, Multithreading and Animation, Abstract Windowing Toolkit, Swings and Event Handling.	8
2	Java Database Connectivity (JDBC): JDBC Product, Types of Drivers, Two- Tier Client/Server Model, Three-Tier Client/Server Model, Basic Steps of JDBC, Creating and Executing SQL Statement, The Result Set Object, Working with Database MetaData Interface.	6
3	Java Servlets: Servlet Interaction & Advanced Servlets, Life cycle of Servlet, Java Servlet Development Kit, Javax.servlet package, Reading Servlet Parameters, Reading Initialization Parameters, The javax.servlet.http Package, Handling HTTP. Java Server Pages (JSP): JSP Technologies, Understanding the Client-Server Model, Understanding Web server software, Configuring the JSP Server, Handling JSP Errors, JSP Translation Time Errors, JSP Request Time Errors, Creating a JSP Error Page, JSP with Java Bean.	10
4	Remote Method Invocation (RMI): RMI Architecture, Designing RMI application, Executing RMI application.	4
	Total Lectures	**28**

Approved in Academic Council held on 28 July 2021

Suggested Text Book(s):

1. Java the Complete Reference, ninth edition by Herbert Schild, Publisher: McGraw Hills
2. Advanced Java Programming by Uttam K. Roy, Publisher: Oxford University Press

Suggested Reference Book(s):

1. Head First Servlets and JSP by Bryan Basham, Kathy Sierra & Bert Bates, Publisher: O'Reilly Media

Other useful resource(s):

1 Link to NPTEL course contents:

i. http://www.nptelvideos.com/java/java_video_lectures_tutorials.php

ii. https://nptel.ac.in/courses/106105084/28

2 Link to topics related to course:

i. https://www.javatpoint.com/

ii. https://www.tutorialspoint.com/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes (Java Programming)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	2	2	1	2	2	3	1	3	1	3	**2.2**
CO-2	3	3	2	2	1	2	1	3	2	2	3	3	**2.3**
CO-3	3	3	2	3	3	3	1	3	2	3	3	2	**2.6**
CO-4	3	3	2	3	3	3	1	3	2	3	3	2	**2.6**
Average	**3**	**3**	**2**	**2.5**	**2**	**2.5**	**1.3**	**3**	**1.8**	**2.8**	**2.5**	**2.5**	

Information Theory and Coding

COURSE CODE: 18B1WCI531

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Basic Probability Theory

Course Objectives:

1. Understand and appreciate how information theory is concerned with the fundamental limits of communication i.e. limit to data compression, and reliable communication over a noisy channel
2. Understand how coding theory is concerned with techniques to realize the limits specified by information theory, and learn the techniques of source coding and channel coding.
3. Get an idea of the broad areas where information theory is used i.e. in statistics, data analysis, cryptography, etc.,
4. Identify how development of information theory and coding theory has been crucial to the development of communications.

Course Outcomes:

S. NO	Course Outcomes	Level of Attainment
CO-1	Understand the basics of information theory and how it is concerned with the fundamental limits of communication	Familiarity
CO-2	Understand the concept of coding and compression techniques	Familiarity
CO-3	Design applications with error control	Usage
CO-4	Use Compression And Decompression Techniques.	Usage
CO-5	Construct efficient codes for data on imperfect communication channels.	Assessment
CO-6	Apply The Concepts Of Multimedia Communication	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Information Entropy Fundamentals** Uncertainty, information and entropy – source coding theorem – huffman coding – shannonfano coding – discrete memory less channels – channel capacity – channel coding theorem – channel capacity theorem.	5
2	**Data And Voice Coding** Differential pulse code modulation – adaptive differential pulse code modulation – adaptive subband coding – delta modulation – adaptive delta modulation – coding of speech signal at low bit rates (vocoders, lpc).	7
3	**Error Control Coding** Linear block codes – syndrome decoding – minimum distance consideration – cyclic codes – generator polynomial – parity check polynomial – encoder for cyclic codes – calculation of syndrome – convolutional codes.	7
4	**Compression Techniques** Adaptive Huffman Coding, Arithmetic Coding, LZW algorithm – Audio: Perceptual	7

	coding, Masking techniques, Psychoacoustic model, MEG Audio layers I,II,III, Dolby AC3 - Speech: Channel Vocoder, Linear Predictive Coding	
5	**Image And Video Coding** Image and Video Formats – GIF, TIFF, SIF, CIF, QCIF – Image compression: READ, JPEG – Video Compression: Principles- I,B,P frames, Motion estimation, Motion compensation, H.261, MPEG standard	6
	Total Lectures	**32**

Suggested Text Book(s):

1. Cover, Thomas M., and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
2. Simon Haykin, “Communication Systems”, 4th Edition, John Wiley And Sons, 2001.
3. Fred Halsall, “Multimedia Communications, Applications Networks Protocols And Standards”, Pearson Education, Asia 2002

Suggested Reference Book(s):

1. Bose, Ranjan. *Information theory, coding and cryptography*. Tata McGraw-Hill Education, 2008.
2. Mark Nelson, “Data Compression Book”, BPB Publication 1992.
3. Watkinson J, “Compression in Video And Audio”, Focal Press, London, 1995.

Other useful resource(s):

1. Link to NPTEL course contents:https://nptel.ac.in/courses/117101053/
2. Link to topics related to course:
   i. http://chamilo2.grenet.fr/inp/courses/PHELMA4PMSTHI9/document/Info_Th_ChI-II-III.pdf?cidReq=PHELMA4PMSTHI9&id_session=0&gidReq=0&origin=

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes(Information Theory and Coding)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	2	1	2	2	2	1	2	3	3	2	**2.2**
CO-2	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
CO-3	3	3	2	1	1	3	3	3	3	3	1	1	**2.3**
CO-4	3	2	2	1	2	2	2	2	1	2	1	2	**1.8**
CO-5	3	2	2	1	2	3	3	2	1	3	2	1	**2.1**
CO-6	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
Average	**3**	**2.7**	**2.2**	**1.2**	**1.7**	**2.5**	**2.7**	**1.8**	**2**	**2.5**	**2**	**1.5**	

Data Compression Lab

COURSE CODE: 18B1WCI572
COURSE CREDITS: 1
CORE/ELECTIVE: ELECTIVE
L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. To provide students with contemporary knowledge in Data Compression and Coding.
2. To equip students with skills to analyze and evaluate different Data Compression and Coding methods.
3. Analyze the operation of a range of commonly used Coding and Compression techniques
4. Identify the basic software and hardware tools used for data compression.
5. Identify what new trends and what new possibilities of data compression are available.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO-1	To understand the statistical basis for and performance metrics for lossless compression.	Familiarity
CO-2	To understand the conceptual basis for commonly used lossless compression techniques	Assessment
CO-3	To understand how to use and evaluate several readily available implementations of those techniques	Assessment
CO-4	To understand the structural basis for and performance metrics for commonly used lossy compression techniques	Assessment
CO-5	To understand the conceptual basis for commonly used lossy compression techniques	Assessment
CO-6	To implement graph theory in compression methodologies for images in MATLAB	Usage
CO-7	To understand image compression techniques‟ case studies	Familiarity

List of Experiments:

S. No	**Description**	**Hours**
1	Implementing a Huffman decoder	**2**
2	Implementing a Huffman encoder	**2**
3	Compressing a large file	**2**
4	Implementation of Run-Length Encoding	**2**
5	Implementation of RLE Text Compression	**2**
6	Implementation of RLE Image Compression	**2**
7	Implementation of Move-toFront Coding	**2**
8	Implementation of Scalar Quantization	**2**
9	Implementing Variable-Size Codes	**2**
10	Implementing Prefix Codes	**2**

Approved in Academic Council held on 28 July 2021

11	Implementing Golomb Codes	**2**
12	Implementing The Kraft-MacMillan Inequality	**2**
13	Implementing The Counting Argument	**2**
14	Implementing Shannon-Fano Coding	**2**
Total Lab hours		**28**

Minor Project(s) – (Only for 2 credit lab)

1. Image Compression using Filtering Techniques.
2. Image Decopression using Wavelets transform.
3. Compressing dehazed images using HE, Fattal method
4. Underwater Image compression using Wavelets and equalization
5. Underwater Panoramic Image compression using mosaicking techniques

Suggested Books/Resources:

1. David Salomon , A Concise Introduction to Data Compression, 1st edition, Springer, 2008
2. Sayood, Khalid, Introduction to Data Compression, 3rd Edition, Morgan Kaufmann, 2006
3. David Salomon, G. Motta, D. Bryan, Data Compression: The Complete Reference, 4nd edition, Springer(2006)
4. D.C. Hankerson, Greg A. Harris , Peter D. Johnson Jr, Introduction to Information Theory and Data Compression, Second Edition, Chapman & Hall/CRC; 2 edition 2003
5. Anderson, J.B. and Mohan, S., Source and Channel Coding, Kluwer, 1991.
6. Gersho, A. and Gray, R.M., Vector Quantization and Signal Compression, Kluwer, 1992.
7. Link to NPTEL course contents: https://nptel.ac.in/courses/108104098/41
8. Link to topics related to course:
   i. https://london.ac.uk/sites/default/files/study-guides/data-compression.pdf
   ii. http://www.ws.binghamton.edu/fowler/fowler%20personal%20page/EE523.htm
   iii. http://www.nhu.edu.tw/~chun/CS-ch15-Data%20Compression.pdf
   iv. http://apachetechnology.in/ati/www/KC/dw/Saloman%20-%20Data_Compression_Complete_Reference.pdf

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	1	2	3	2	3	3	3	3	2	**2.6**
CO-2	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-3	3	3	3	3	3	3	3	3	3	3	3	3	**3**

CO-4	3	3	2	3	3	3	2	1	1	3	3	3	**2.5**
CO-5	3	3	3	1	2	3	2	3	3	3	3	1	**2.5**
CO-6	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-7	3	3	3	3	3	3	3	3	3	3	3	3	**3**
Average	**2.7**	**2.7**	**2.6**	**2.4**	**2.7**	**3**	**2.3**	**2.7**	**2.7**	**3**	**2.4**	**2.6**	

Principles of Programming Languages Lab

COURSE CODE: 18B1WCI573

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Learn syntax, semantics and create Functions in Principles of Programming Languages.
2. Recite the high points of programming language history.
3. Read the central formalisms used in the description of programming languages.
4. Assess programming languages critically and in a scientific manner;
5. Analyze the principles of an imperative, functional, object oriented or logic oriented programming language.

Course Outcomes:

S. NO	Course Outcomes	Level of Attainment
CO-1	To learn major programming paradigms and techniques involved in design and implementation of modern programming languages. To learn the structure of a compiler and interpretation. To learn syntax and semantic of programming language.	Familiarity
CO-2	To learn the structured and object oriented programming paradigm.	Familiarity
CO-3	To different programming paradigm to improving the clarity, quality, and development time of a program (structured programming). To learn Haskell (an advanced purely-functional programming style and lambda calculus (for variable binding and substitution).	Assessment
CO-4	To learn the concurrency in programming languages, Exception handling and Scripting languages	Familiarity

List of Experiments:

S. No	Description	Hours
1	1. To understand the value of records in a programming language, write a small program in a C-based language that uses an array of structs that store student information, including name, age, GPA as a float, and grade level as a string (e.g., “freshmen,” etc.). Also, write the same program in the same language without using structs. 2. To understand the value of recursion in a programming language, write a program that implements quicksort, first using recursion and then without recursion.	**2**

	1. Write a BNF description of the Boolean expressions of Java, including the three operators && || and ! and the relational expressions.
	2. Prove the following program is correct: a. {n >
	0}
	b. count = n;
	c. sum = 0;
	d. while
	e. count <> 0
	f. do
	g. sum = sum + count;
	h. count = count - 1;
	i. end
	j. {sum = 1 + 2 + ...+ n}
		2

3	**1.** Implement and test the LR parsing algorithm [Hint Section 4.5.3 page 214 book concepts-of-programming-languages]	**2**

4	**1.** Write a C function that includes the following sequence of statements:
	x = 21;
	int x;
	x = 42;
	Run the program and explain the results. Rewrite the same code in C++ and Java and compare the results.
	
	2. Write three functions in C or C++: one that declares a large array statically, one that declares the same large array on the stack, and one that creates the same large array from the heap. Call each of the subprograms a large number of times (at least 100,000) and output the time required by each. Explain the results
		2

5	**1.** Write a program in the language of your choice that behaves differently if the language used name equivalence than if it used structural equivalence.. **2.** Write a Java program that exposes Java‟‟s rule for operand evaluation order when one of the operands is a method call.	**2**
6	**1.** Rewrite the following code segment using a multiple-selection statement in the following languages: *if* *((k == 1) || (k == 2)) j = 2 * k - 1 if* *((k == 3) || (k == 5)) j = 3 * k + 1 if* *(k == 4) j = 4 * k - 1 if* *((k == 6) || (k == 7) || (k == 8)) j = k – 2* *a. Fortran 95 (you'll probably need to look this one up)* *b. Ada* *c. C, C++, Java, or C#* *d. Python* *e. Ruby* Assume all variables are integer type. Discuss the relative merits of the use of these languages for this particular code.	**2**
7	**1.** Write a Perl program that passes by reference a literal to a subprogram, which attempts to change the parameter. Given the	**2**

	overall design philosophy of Perl, explain the results. **2.** Write a program in some language that has both static and stack dynamic local variables in subprograms. Create six large (at least 100*100) matrices in the subprogram three static and three stack dynamic. Fill two of the static matrices and two of the stack-dynamic matrices with random numbers in the range of 1 to 100. The code in the subprogram must perform a large number of matrix multiplication operations on the static matrices and time the process. Then it must repeat this with the stack-dynamic matrices. Compare and explain the results **3.** Write a generic Ada function that takes an array of generic elements and a scalar of the same type as the array elements. The type of the array elements and the scalar is the generic parameter. The subscripts of the array are positive integers. The function must search the given array for the given scalar and return the subscript of the scalar in the array. If the scalar is not in the array, the function must return –1 . Instantiate the function for Integer and Float types and test both.	
8	**1.** Write a program that includes two subprograms, one that takes a single parameter and performs some simple operation on that parameter and one that takes 20 parameters and uses all of the parameters, but only for one simple operation. The main program must call these two subprograms a large number of times. Include in the program timing code to output the run time of the calls to each of the two subprograms. Run the program on a RISC machine and on a CISC machine and compare the ratios of the time required by the two subprograms. Based on the results, what can you say about the speed of parameter passing on the two machines?	**2**
9	**1.** Write an abstract data type for queues whose elements store 10- character names. The queue elements must be dynamically allocated from the heap. Queue operations are enqueue, dequeue, and empty. Use either Ada, C++, Java, C#, or Ruby. **2.** Write an abstract data type for rational numbers (a numerator and a denominator). Include a constructor and methods for getting the numerator, getting the denominator, addition, subtraction, multiplication, division, equality testing, and display. Use Java, C#, C++, Ada, or Ruby.	**2**
10	**1.** Design and implement a C++ program that defines a base class A, which has a subclass B, which itself has a subclass C. The A class must implement a method, which is overridden in both B and C. You must also write a test class that instantiates A, B, and C and includes three calls to the method. One of the calls must be statically bound to A‟s method. One call must be dynamically bound to B‟s method, and one must be dynamically bound to C‟s method. All of the method calls must be through a pointer to class A. **2.** The reader-writer problem can be stated as follows: A shared memory location can be concurrently read by any number of tasks, but when a task must write to the shared memory location, it must have exclusive access. Write a Java program for the reader-writer problem.	**2**
11	**1.** Write a Java program that inputs a list of integer values in the range of - 100 to 100 from the keyboard and computes the sum of the squares of the input values. This program must use exception handling to ensure that the input values are in range and are legal	**2**

	integers, to handle the error of the sum of the squares becoming larger than a standard Integer variable can store, and to detect end- of-file and use it to cause the output of the result. In the case of overflow of the sum, an error message must be printed and the program terminated.	
12	**1.** Write a Scheme function that computes the real roots of a given quadratic equation. If the roots are complex, the function must display a message indicating that. This function must use an *IF* function. The three parameters to the function are the three coefficients of the qua dratic equation.	**2**
13	**1.** Write a Scheme function that takes a list as a parameter and returns a list identical to the parameter list except with the second top-level element removed. If the given list does not have two elements, the function should return (). Rewrite the following Scheme function as a tail-recursive function: *(DEFINE (doit n) (IF (= n* *0)* *0* *(+ n (doit (− n 1)))* *))*	**2**
14	***1.*** Using the structures *parent(X, Y)* , *male(X)* , and *female(X)* , write a structure that defines *mother(X, Y)* **2.** Write a Prolog program that succeeds if the intersection of two given list parameters is empty. **3.** Write a Prolog program that implements quicksort.	**2**
Total Lab Hours		**28**

Suggested Books/Resources:

1. https://cs444pnu1.files.wordpress.com/2014/02/concepts-of-programming-languages-10th-sebesta.pdf
2. Principles of Programming Languages: Design, Evaluation, and Implementation 3rd Edition by Bruce J. MacLennan
3. Programming Languages: Concepts & Constructs, 2nd Edition by Ravi Sethi; Pearson Education Asia
4. Programming Language Principles and Paradigms 2nd Edition, Tucker, Allen B. Michael and Noonan Robert E. , TMH 2007
5. Programming Languages and Paradigms, D. A. Watt., Prentice-Hall,1990
6. Essentials of Programming Languages, Daniel Friedman, Mitchell Wand, and Christopher Haynes, MIT Press (Indian edition - Prentice Hall, India)
7. Concepts of Programming Languages, Robert W. Sebesta, Pearson Education Asia
8. https://nptel.ac.in/courses/106102067/
9. https://www.cs.bgu.ac.il/~ppl162/Assignments

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	3	1	2	3	2	2	1	1	1	3	**2**
CO-2	3	2	3	1	1	3	2	3	2	1	1	3	**2.1**
CO-3	3	2	3	1	1	2	2	2	1	1	1	3	**1.8**
CO-4	3	2	3	2	2	3	2	2	1	1	1	3	**2.1**
Average	**3**	**2**	**3**	**1.3**	**1.5**	**2.8**	**2**	**2.3**	**1.3**	**1**	**1**	**3**	

Java Programming Lab

COURSE CODE: 18B1WCI574

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Using Graphics, Animations and Multithreading for designing Simulation and Game based applications.
2. Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling.
3. Design and develop Web applications
4. Designing distributed applications using Remote Method Invocation (RMI)

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the graphics and animation on the web pages, using Java Applets. To learn and design a full set of Event driven UI widgets and other components, including windows, menus, buttons, checkboxes, text fields, scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT) & Swings	Usage
CO-2	To learn Java Data Base Connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through Java programs.	Usage
CO-3	To learn the server side programming using Servlets and JSP.	Usage
CO-4	To learn the invocation of the remote methods in an application using RMI	Usage

List of Experiments:

S. No.	Description	Hours
1	Design an applet to continuously scroll a string from the right to the left of the applet window.	2
2	Design an applet called Checker in which a black oval moves from a green square to a blue square. To reduce the flickering override the update method to clear only that part of the screen that is changing during the animation.	2
3	Write a swing-based GUI to create a tabbed pane with two tabs: Table and Tree. The first tab contains a table displaying the information of Employee {empid, ename, age}. The second tab contains a tree structure. Insert both JTable and JTree in JScrollPane. JScrollPane has to be inserted at the centre of the BorderLayout manager (with panels). When user clicks on any node of the tree, its path should be displayed on the TextField.	2
4	Consider a table Bank {account_no, customer_name, balance, phone_no, and address}. Write a database application which allows insertion, updation and deletion of records in the Bank table. Print values of all customers whose balance is less than 5,000.	2
	User enters the name and password through an HTML form. Both these parameters are being passed to the Servlet1. The Servlet1 after retrieving the parameters generates a Cookie with username being stored in the cookie. The Servlet1 generates an HTML form, which is linked to the Servlet2. Along with the response object, the Cookie is stored permanently on the client‟s hard disk. The same user then submits the form to invoke the Servlet2. The Servlet2 must have the provision to access and display the	2

5	cookie name. Modify the above program to use the concept of Hidden form field, instead of Cookie.	
6	Write a Servlet to validate the username and password entered by the user. If the username and password are ***„abc' and 'def'*** respectively, the Sevlet should forward the request to ***WelcomeServlet***; otherwise, it should display an error message: "*Incorrect username or password!*"	**2**
7	Write a Servlet that connects to the database (MySql) and creates a table ***emp*** with the following schema: ***emp***{empid char(5) Primary Key, ename varchar2 not null , age number(2) not null, salary number(7, 2) , address varchar2};	**2**
8	• Write a Servlet that connects to the database (MySql), runs a query to retrieve the ***ids*** and ***names*** of the ***emp*** table and displays the list to the client. • Write a Servlet to insert data into a table, using ***executeUpdate()*** and then display the result using ***executeQuery()*** Modify the above ***Servlet*** to insert the data into the ***emp*** table, using ***PreparedStatement*** interface	**2**
9	Write a JSP program to display your brief profile including your image, date of birth, address, and educational qualification. Do proper formatting and coloring of this web page.	**2**
10	Write an online phone directory based application using JSP and JDBC. Use the following table to store the phone details: Directory{phone_no, first_name, last_name, address}. The application should have the search facility on all the fields.	**2**
11	Write a JSP program using Java Bean to register users into a web-site. The registration details of users are kept in USER table with attributes as userid varchar2(10), username varchar2(15), and email_id varchar2(15).	**2**
12	Write a distributed application using RMI where a user running an application on the client‟s JVM invokes a remote method String getName(String) running on the server‟s JVM. After invoking this method, the name entered by the user is displayed on the client‟s application.	**2**
13	Write an RMI application where client sends empno and server returns corresponding salary by querying database	**2**
14	Write one RMI application for banking to perform the deposit and withdrawal operations by client.	**2**
Total Lab hours		**28**

Suggested Books/Resources:

1. Java the Complete Reference, ninth edition by Herbert Schild, Publisher: McGraw Hills
2. Advanced Java Programming by Uttam K. Roy, Publisher: Oxford University Press
3. Head First Servlets and JSP by Bryan Basham, Kathy Sierra & Bert Bates, Publisher: O'Reilly Media
4. Link to topics related to course:
   - **i.** http://www.nptelvideos.com/java/java_video_lectures_tutorials.php
   - **ii.** https://nptel.ac.in/courses/106105084/28
   - **iii.** https://www.javatpoint.com/
   - **iv.** https://www.tutorialspoint.com/

Evaluation Scheme

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks

4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	2	2	1	1	2	3	1	3	2	3	**2.2**
CO-2	3	3	2	2	1	1	1	3	2	2	3	3	**2.2**
CO-3	3	3	2	3	3	3	1	3	2	3	3	2	**2.6**
CO-4	3	3	2	3	3	3	1	3	2	3	3	2	**2.6**
Average	**3**	**3**	**2**	**2.5**	**2**	**2**	**1.3**	**3**	**1.8**	**2.8**	**2.8**	**2.5**	

Information Theory and Coding Lab

COURSE CODE: 18B1WCI571

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. To develop skills for designing programs on different coding techniques.
2. Understand how coding theory is concerned with techniques to realize the limits specified by information theory, and learn the techniques of source coding and channel coding.
3. Get an idea of the broad areas where information theory is used i.e. in statistics, data analysis, cryptography, etc.,
4. Identify how development of information theory and coding theory has been crucial to the development of communications.

Course Outcomes:

S. NO	Course Outcomes	Level of Attainment
CO-1	Understand the basics of information theory and how it is concerned with the fundamental limits of communication	Familiarity
CO-2	Design and implement concept of coding and compression techniques	Familiarity
CO-3	Design applications with error control	Usage
CO-4	Design and implement Compression And Decompression Techniques.	Usage
CO-5	Construct and implement efficient codes for data on imperfect communication channels.	Assessment
CO-6	Design and use the concepts Of Multimedia Communication	Assessment

List of Experiments:

S. No.	Description	Hours
1	Write a program for determination of various entropies and mutual information of a given channel. Test various types of channel such as a) Noise free channel. b) Error free channel c) Binary symmetric channel d) Noisy channel Compare channel capacity of above channels.	**2**
2	Write a Program to implement an algorithm for Determination of Entropy, Information, and Information Rate.	**2**
3	Write a program for generation and evaluation of variable length source coding using Shannon – Fano coding and decoding	**2**
4	Write a program for generation and evaluation of variable length source coding using Huffman Coding and decoding	**2**
5	Write a Program for coding & decoding of Linear block codes.	**2**
6	Write a Program for coding & decoding of Cyclic codes.	**2**
7	Write a program for coding and decoding of Convolutional codes.	**2**
8	Write a program for coding and decoding of BCH codes.	**2**
9	Write a program for coding and decoding of RS codes.	**2**

10	Write a program to study performance of a coded and uncoded communication system(Calculate the error probability)	**2**
11	Write a simulation program to implement source coding and channel coding for transmitting a text file.	**2**
12	Write a program to implement adaptive Huffman coding to compress the textual data.	**2**
13	Write a program to implement LZW technique to compress the textual data.	**2**
14	Write a program to implement perceptual coding technique to compress audio data.	**2**
Total Lab Hours		**28**

Suggested Books/Resources:

1. Cover, Thomas M., and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
2. Simon Haykin, "Communication Systems", 4th Edition, John Wiley And Sons, 2001.
3. Fred Halsall, "Multimedia Communications, Applications Networks Protocols And Standards", Pearson Education, Asia 2002
4. Bose, Ranjan. *Information theory, coding and cryptography*. Tata McGraw-Hill Education, 2008.
5. Mark Nelson, "Data Compression Book", BPB Publication 1992.
6. Watkinson J, "Compression in Video And Audio", Focal Press, London, 1995.
7. Link to NPTEL course contents:https://nptel.ac.in/courses/117101053/
8. Link to topics related to course:
   i. http://chamilo2.grenet.fr/inp/courses/PHELMA4PMSTHI9/document/Info_Th_ChI-II-III.pdf?cidReq=PHELMA4PMSTHI9&id_session=0&gidReq=0&origin=

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**Average**
CO-1	3	3	2	2	2	2	2	1	2	3	3	1	**2.2**
CO-2	3	1	3	2	2	2	3	2	2	3	2	2	**2.3**
CO-3	3	3	2	1	1	3	3	3	3	3	1	1	**2.3**
CO-4	3	2	2	2	2	2	2	2	1	2	1	1	**1.8**
CO-5	3	3	2	1	2	3	3	2	1	2	2	1	**2.1**
CO-6	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
Average	**3**	**2.5**	**2.2**	**1.5**	**1.7**	**2.5**	**2.7**	**1.8**	**2**	**2.7**	**2**	**1.2**	

Compiler Design

COURSE CODE: 18B11CI612

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

: 3-0-0

Pre-requisites: Perquisite for this course is programming languages like C, C++, java and basic knowledge of formal language and automata theory. This is a programming-intensive course. The primary languages used are C and Java and familiarity with them is assumed.

Course Objectives:

1. To understand the basic concept of compilation particular, lexical analyzer, syntax and semantic analysis, code generation and optimization phases of compilation.
2. Ability to create lexical rules and grammars for a programming language.
3. Ability to use flex or similar tools to create a lexical analyzer and Yacc/Biscon tools to create a parser
4. Ability to implement a various parser such as a bottom-up SLR parser without using any compiler-generation tools
5. Ability to implement a various intermediate code generation techniques without using any compiler generation tools
6. Ability to implement various code optimizing techniques without using any compiler-generation tools
7. Ability to implement semantic rules into a parser that performs attribution while parsing
8. Ability to design a compiler for a concise programming language.

Course Outcomes

S. No.	Course Outcomes	Level of Attainment
CO-1	To understand the basic concept of compilation particular, lexical analyzer, syntax and semantic analysis, code generation and optimization phases of compilation	Familiarity
CO-2	Ability to create lexical rules and grammars for a programming language	Familiarity
CO-3	Ability to use flex or similar tools to create a lexical analyzer and Yacc/Biscon tools to create a parser	Computational skills
CO-4	Ability to implement a various parser such as a bottom-up SLR parser without using any compiler-generation tools	Technical skills
CO-5	Ability to implement a various intermediate code generation techniques without using any compiler generation tools	Technical skills
CO-6	Ability to implement various code optimizing techniques without using any compiler-generation tools	Technical skills
CO-7	Ability to implement semantic rules into a parser that performs attribution while parsing	Assessment
CO-8	Ability to design a compiler for a concise programming language	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Compilers, Analysis of Source Program, Phases of Compiler, Issues in a Compiler Structure, Major Data Structures in Compiler, Bootstrapping and Porting	3
	Lexical Analysis: Role of the Lexical Analyzer, Scanning Process- Input Buffering, Specification of Tokens, Recognition of Tokens, Regular Language,	

2	Finite Automata, Regular Expression, Regular Expression to Finite automata, Scanner Generator (lex, flex)	**4**
3	**Syntax Analysis:** Parsing Process, Context-Free Grammar, Handle, Ambiguity, Parse Tree, Push-Down Automata , Top Down Parsing: Basic Concept, Recursive Descent Parsing, First Function, Follow Function, Conflicts-Shift Reduce, Reduce-Reduce, LL(1) Grammar, LL(1) Parser, Error Recovery, Bottom up parsing :Basic Concept, Shift Reduce Parsing, Operator Grammar, Operator Precedence Parsing, Augmented Grammar, LR(0) Item Set, LR(0) Parsing, LR(1) Item Set, SLR(1) Parsing, Canonical LR Parser, LALR parsing, Error Recovery, YACC and BISON	**12**
4	**Semantic Analysis:** Syntax-Direct Definitions, Implementation of Syntax-directed Translators, Bottom-up Evaluation of S-attributed Definitions, L-attributed Definitions, and Top-down Translation	**5**
5	**Intermediate Code Generation:** Postfix Notation, Parse Trees & Syntax Trees, Three Address Code, Quadruple & Triples, Translation of Assignment Statements, Boolean Expressions, Statements That Alter Flow of Control, Postfix Translation, Array References in Arithmetic Expressions, Procedures Call, Declarations and Case Statements	**5**
6	**Code Generation: Issues** in the design of a Code Generator, Target Machine, Run-Time Storage Management, Basic blocks and Flow graphs	**4**
7	**Code Optimization:** Machine-Independent Optimizations, Machine Dependent Optimization, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis	**5**
8	**Run-time environment and Symbol:** Procedure activation, parameter passing, value return, memory allocation, and scope, Symbol Table, Definition, Structure, Types, symbol attributes and management	**4**
Total Lectures		**42**

Suggested Text Book(s):

1. Compilers: Principles, Techniques, and Tools by Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman, Pearson Publisher
2. Compiler Construction, Kenneth C. Louden, 582 pages
3. Modern Compiler Implementation in C, Andrew W. Appel, 544 pages

Suggested Reference Book(s):

1. http://ecomputernotes.com/compiler-design
2. http://www.diku.dk/~torbenm/Basics/basics_lulu2.pdf

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Compiler Design)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	2	2	1	2	3	3	2	1	3	2	1	**2.1**
CO-2	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
CO-3	3	3	2	1	2	2	2	1	2	3	3	2	**2.2**
CO-4	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
CO-5	3	3	3	2	2	2	2	2	1	3	2	1	**2.2**
CO-6	3	3	3	3	2	3	3	2	2	3	2	2	**2.6**
CO-7	3	3	3	3	2	3	3	1	2	3	2	2	**2.5**
CO-8	3	3	3	3	2	3	3	2	2	2	3	2	**2.6**
Average	**3**	**2.9**	**2.6**	**2**	**1.9**	**2.6**	**2.8**	**1.6**	**1.9**	**2.6**	**2.4**	**1.6**	

Computer Networks

COURSE CODE: 18B11CI611

COURSE CREDITS: 3

CORE/ELECTIVE: CORE

: 3-0-0

Pre-requisites: None

Course Objectives:

1. The course introduces the concepts and fundamental design principles of modern computer networking, focusing on the Internet"s architecture and protocols.
2. The course introduces the concepts of data and computer communications, computer network introduction and its applications in our real life.
3. Reference models such as OSI and TCP/IP and its way toward the physical layer concepts, data link layer and its protocols, multiple access protocols.
4. Provide students Network layer and its different routing protocols, the concepts/design of IP addressing.
5. To provide students Transport layer and its protocols such as TCP, UDP and SCTP to application layer and its protocol such as HTTP, FTP, SMTP and DNS.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the basic concepts and terminology in computer networks	Familiarity
CO-2	To learn about the layered models in computer networks and different types of network topologies and protocols	Assessment
CO-3	To learn about the data link layer and MAC layer protocols and related issues	Assessment
CO-4	To learn concepts associated with subnetting and routing mechanisms. Understand network industry standards such as: Routing Protocols, Address Resolution and Reverse Address Resolution Protocols, IP Addresses and Subnetting, MAC Addressing.	Assessment
CO-5	To learn about the transport layer protocols and related issues	Assessment
CO-6	You will learn about the session, presentation and application layers protocols.	Usage
CO-7	Further, to learn about the some advanced topics in networks such as Cryptographic algorithms, Network security and management, and concepts of wireless networks	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Computer Networks**: Network Software Architecture: layers and protocols, OSI vs. TCP, Network Model, Connection Oriented and Connectionless services, Network Topology, Delay.	**4**
2	**Physical Layer:** Transmission Terminology, Analog and Digital Signal, Transmission Impairments, Transmission Media, Modulation, Switching Techniques, Multiplexing Techniques	**8**

3	**Data Link Layer:** Introduction and services to Data Link layer, Error detection and Correction techniques, Bit and Byte stuffing, Bit/Byte oriented protocol, Flow Control Mechanism, Multiple access protocol, Ethernet, Hubs and switches, Router and Gateways.	**8**
4	**Network Layer:** Network service model, Virtual circuit and Datagram networks, Logical Addressing and Sub-netting, Internet protocol: IPv4 and IPv6, ARP vs RARP, DHCP, Routing algorithms and standards, Internetworking, The network layer in the internet, Broadcast and multicast routing, Congestion Control Algorithms,	**8**
5	**Transport Layer:** Transport layer services and principles, End-to-end protocols: Issues and services, Multiplexing and De-multiplexing, Connectionless transport: UDP, Principles of reliable data transfer, Connection-oriented Transport: TCP, SCTP, Principles of congestion control, TCP Congestion Control, Quality of services.	**8**
6	**Application Layer:** Principle of application layer protocols, WWW and HTTP, FTP, Telnet, SMTP, DNS etc.	**3**
7	**Some Advanced Topics:** Symmetric-key algorithms, Public key algorithms, RSA, Digital Signatures, Communication security, authentication protocols, Web security, Wireless LAN, Mobile IP, Introduction to Multimedia networking, Network management.	**3**
Total lectures		**42**

Suggested Text Book(s):

1. Andrew S. Tanenbaum, “Computer Networks” 4th Edition PHI
2. James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach Featuring the Internet” 3rd Edition Pearson Education

Suggested Reference Book(s):

1. UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998, ISBN 0-13-490012-X.
2. Arnold Robbins, “UNIX in a Nutshell”, O‟Reilly 4th Edition
3. David I. Schawartz, “Introduction to UNIX”, Prentice Hall, Second Edition
4. BEHROUZ a. Forouzan and Richard F. Gilberg, “UNIX and Shell Programming: A Textbook”
5. NS Simulator for Beginner‟s, Lecture notes Univ. de Los Andes, France.
6. Angela Orebaugh, Gilbert Ramirez,Josh Burke, Larry Pesce, Joshua Wright, Greg Morris, “Wireshark & Ethereal Network Protocol Analyzer Toolkit”, Syngress Publishing, Inc.

Other useful resource(s):

3. Link to topics related to course:

i. https://www.coursera.org/specializations/networking-basics
ii. https://nptel.ac.in/courses/106105080/
iii. https://swayam.gov.in/course/4066-computer-networks

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Computer Networks)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
CO-7	3	3	3	2	2	3	3	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.6**	**2.9**	**2.1**	**2**	**2.4**	**3**	**1**	**3**	

Compiler Design Lab

COURSE CODE: 18B17CI672

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: An understanding in Theory of computation, Introduction to any programming language (Preferably, C)

Course Objectives:

1. The lab course provides the complete description about inner working of a compiler.
2. The main focus is on the design of compilers and optimization techniques.
3. The course also aims to convey the language specifications, use of regular expressions and context free grammars behind the design of compiler.
4. It builds an understand ability of various parsing techniques like predictive parsing, LR parsing, LALR parsing. It also focuses on the design of Compiler writing tools.

Course Outcomes:

S. No.	**Course Outcomes**	**Level of Attainment**
CO-1	Gain an in-depth understanding of the principles underlying the design	Familiarity
CO-2	Construction of compilers	Familiarity
CO-3	Functioning of Compiler writing tools	Computational skills
CO-4	Building various parsing techniques	Technical skills

List of Experiments:

S. No.	**Description**	**Hours**
1	a. Write a program to read and translate integers into numbers. e.g. 1=ONE 12 = ONE TWO 856 = EIGHT FIVE SIX Generate an error if the number of digits is more than 3 b. Write a program to convert infix notation to postfix notation.	4
2	1. Implement a DFA which simulates the regular expression a + (aa)*b. 2. The following rules define the translation of an English word into pig Latin: a) If the word begins with a nonempty string of consonants, move the initial consonant string to the back of the word and add the suffix AY; e.g., pig comes igpay. b) If the word begins with a vowel, add the suffix YAY; e.g., owl becomes owlyay.	8

	c) U following a Q is a consonant. d) Y at the beginning of a word is a vowel if it is not followed by a vowel. e) One-letter words are not changed. Write a C program to generate pigLatin from an English word.	
3	Implementation of Lexical analysis	4
4	Program for computation of FIRST AND FOLLOW of non-terminals.	4
5	Write a program to check whether a grammar is left recursive or not, if it is remove left recursion.	4
6.	Implementation of Predictive Parsing Table Construction	6
7.	Implementation of Shift Reduce Parsing	6
8.	Implementation of Operator Precedence Parsing	6
9.	Implementation of LR Parsing	4
10	Intermediate Code Generation	4
11	Implementation of Code Generation	4
Total Lab hours		**56**

Suggested Books/Resources:

1. Compilers : Principles, techniques and tools, Aho, Sethi and Ullman
2. Compiler design in C, Holub
3. Advanced compiler design and implementation, Muchnick, Morgan and Kauffman

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	2	1	2	3	3	2	1	3	2	1	**2.1**
CO-2	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
CO-3	3	3	2	1	2	2	2	1	3	3	3	2	**2.3**
CO-4	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
Average	**3**	**2.8**	**2.3**	**1.3**	**1.8**	**2.5**	**2.8**	**1.5**	**2.3**	**2.5**	**2.5**	**1.5**	

Computer Networks Lab

COURSE CODE: 18B17CI671

COURSE CREDITS: 2

CORE/ELECTIVE: CORE

: 0-0-4

Pre-requisite: None

Course Objectives:

1. To implement important computer networking protocols in a high – level programming language.
2. To understand the working principle of various communication protocols.
3. To analyze the various routing algorithms.
4. To know the concept of data transfer between nodes.
5. To become acquainted with socket programming and some of the important computer networking tools.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	To understand the working concepts of Networking and inter – networking Devices.	Familiarity
CO-2	To understand the concepts of different shortest path algorithms.	Assessment
CO-3	To understand different error detection and correction techniques/algorithms.	Assessment
CO-4	To understand Flow control techniques/algorithms.	Assessment
CO-5	To understand the concepts of client – server interaction using connection oriented and connectionless protocols.	Assessment
CO-6	To understand the proficiency in Traffic Shaping Algorithms.	Usage
CO-7	You shall be exposed to working of encryption and decryption algorithms.	Familiarity

List of Experiments:

S. No.	Description	Hours
1	Representation of a computer network using matrix representation of a graph	4
2	Finding shortest path between any two nodes in a computer network using Dikjstra‟s shortest path algorithm	4
3	Finding shortest path between any two nodes in a computer network using Prim‟s shortest path algorithm	4
4	Study of network troubleshooting using Ping and Traceroute commands	4
5	Study of various networking and inter – networking devices	4
6	Implementation of CRC generator and checker algorithm in C / C++ / Java	4
7	Implementation of Hamming code algorithm in C / C++ / Java	4
8	Study of client – server programming using sockets in a UNIX / Linux and Windows environment	4
9	Implementing client – server program using TCP / UDP sockets	4
10	Implementation of Stop – and – Wait protocol in C / C++ / Java in a client – server environment using sockets	4

11	Implementation of Sliding Window protocol in C / C++ / Java in a client – server environment using sockets	**4**
12	Implementation of encryption algorithm converting plain text to cipher text using C / C++ / Java	**4**
13	Design and implement Traffic Shaping Algorithms: a. Leaky Bucket b. Token Bucket	**4**
14	Implementation of chat system	**4**
15	Allocation of Mini projects	**4**
Total Lab hours		**56**

Suggested Books/Resources:

1. James F. Kurose, Keith W. Ross, "Computer Networking: A Top-Down Approach Featuring the Internet" 3rd Edition Pearson Education
2. Andrew S. Tanenbaum, "Computer Networks" 4th Edition PHI
3. UNIX Network Programming, Volume 1, Second Edition: Networking APIs: Sockets and XTI, Prentice Hall, 1998, ISBN 0-13-490012-X.
4. Arnold Robbins, "UNIX in a Nutshell", O"Reilly 4th Edition
5. David I. Schawartz, "Introduction to UNIX", Prentice Hall, Second Edition
6. BEHROUZ a. Forouzan and Richard F. Gilberg, "UNIX and Shell Programming: A Textbook"
7. NS Simulator for Beginner"s, Lecture notes Univ. de Los Andes, France.
8. Link to topics related to course:
   i. https://www.coursera.org/specializations/networking-basics
   ii. https://nptel.ac.in/courses/106105080/
   iii. https://swayam.gov.in/course/4066-computer-networks

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	3	3	1	1	3	3	3	1	3	**2.5**
CO-2	2	2	2	2	2	1	1	1	2		1	2	**1.6**
CO-3	3	3	3	3	3	1	1	1	3	1	1	3	**2.2**
CO-4	2	2	2	2	2	2	1	1	2		2	2	**1.8**
CO-5	2	2	2	2	2	3	1	1	2	2	2	2	**1.9**
CO-6	2	2	2	2	2	2	1	2	2	2	2	2	**1.9**
CO-7													**1.9**

	2	2	2	2	2	2	1	2	2	2	2	2	
Average	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**1.7**	**1**	**1.6**	**2.3**	**2**	**1.6**	**2.3**	

DETAILED COURSE DESCRIPTIONS

ELECTIVE - II

Software Testing Fundamentals

COURSE CODE: 18B1WCI633

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: C/C++, Python, Eclipse, Netbeans

Course Objectives:

1. Employ correct testing terminology throughout the testing process
2. Execute specific software tests with well-defined objectives and targets.
3. Modelling techniques: UML: FSM and State charts, combinatorial design; and others.
4. Apply various testing techniques, including domain, code, fault, usage and model-based.
5. Perform a complete testing process, taking into account practical considerations**.**

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Investigate the reason for bugs and analyze the principles in software testing to prevent and remove bugs.	Familiarity
CO-2	Implement various test processes for quality improvement.	Assessment
CO-3	Design test planning and manage the test process.	Assessment
CO-4	Apply the software testing techniques in commercial environment.	Assessment
CO-5	Design test adequacy assessment and enhancement criteria.	Assessment
CO-6	Use practical knowledge of a variety of ways to test software and an understanding of some of the tradeoffs between testing techniques.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Error, Fault, Failure, Test automation and the importance of testing, Developer and tester as two roles, Principles of Testing, ETVX Model, Testing Maturity Model, V-Model, Software quality, Testing and debugging (preparing, Constructing, Executing, Specifying, Assessing a test plan), test Generation Strategies, Types of testing and Classifiers C1, C2, C3, C4, and C5, Static testing **Preliminaries mathematical:** Predicates and Boolean Expressions, Control Flow Graph, Program Dependence Graph, Strings languages and regular expressions,	**6**
2	**Test Generation:**	**10**
	a) From Domain Partitioning: The test selection problem, Equivalence partitioning, Boundary value analysis, Category- partition method, Cause-effect graphing. **b) From Finite State Models**: Finite State machines, Conformance testing, A Fault model, Characterization Set, The w-Method, The partial W-methos. **c) From Combinatorial design**: Combinatorial designs, A combinatorial test design process, Fault model, Latin Squares, Mutually orthogonal Latin squares, Pairwise designs: binary factors, Pairwise design: multi-valued factors, Orthogonal	

	Arrays.	
3	**Test Adequacy Assessment and Enhancement:** **a) Using Control flow:** Test adequacy basics, adequacy criteria based on control flow – Statement coverage, Decision coverage, condition coverage, MCC, LCSAJ, basis path coverage, **b) Using data Flow:** Definitions, C-use, p-use, Data flow graphs, du-path, dc-path, c-use coverage, p-use coverage, All-use coverage, k-dr chain coverage. **c) Using Mutation:** Mutation and Mutants, Test Assessment using mutation, Mutation operators, Founding principles of mutation testing, Equivalent mutants, Fault detection using mutation, Types of mutants.	**8**
4	**Phases of testing I:** **Regression testing:** Regression test process, Regression test selection, Selecting regression tests, test selection using execution trace, test selection using dynamic slicing	**4**
5	**Phases of testing II:** Unit Testing, Integration Testing, System testing, Acceptance testing.	**4**
Total lectures		**32**

Suggested Text Book(s):

1. "Foundations of Software testing," 2nd edition by Aditya P mathur, Pearson 2013
2. "Practical Software testing," 8th edition by Ilene Burnstein, Springer 2010

Suggested Reference Book(s):

1. Paul C. Jorgensen, Software testing: a Craft‟s man approach, CRC Press
2. Srinivasan Desikan and G. Ramesh, Software Testing: Principles and Practices, Pearson Education
3. Glenford Myers, "The Art of Software Testing", John Wiley & Sons Inc., New York, 1979.
4. Cem Kaner, Jack Falk, Nguyen Quoc, "Testing Computer Software", Second Edition, Van Nostrand Reinhold, New York, 1993.
5. Boris Beizer, "Software Testing Techniques", Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
6. Roger S. Pressman, "Software Engineering – A Practitioner‟s Approach", Fifth Edition, McGraw- Hill International Edition, New Delhi, 2001.
7. Boris Beizer, "Black-Box Testing – Techniques for Functional Testing of Software and Systems", John Wiley & Sons Inc., New York, 1995.
8. K.K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International Publishers, New Delhi, 2003.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105150/
2. Link to topics related to course:
   i. https://www.guru99.com/software-testing.html
   ii. https://www.inf.ed.ac.uk/teaching/courses/st/2011-12/Resource-folder/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Software Testing Fundamentals)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Machine Learning

COURSE CODE: 18B1WCI634

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisite: None

Course Objectives:

1. Have a good understanding of the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
2. Have an understanding of the strengths and weaknesses of many popular machine learning approaches.
3. Appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
4. Be able to design and implement various machine learning algorithms in a range of real-world applications.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the basic concepts and terminology in machine learning	Familiarity
CO-2	To learn about the definition of learning systems, their goals and applications in machine learning	Familiarity
CO-3	To understand concepts associated with classification and experimental evaluation of classification algorithms	Assessment
CO-4	To learn concepts associated with decision trees and experimental evaluation of classification algorithms	Assessment
CO-5	To learn about instance-based learning, clustering and unsupervised learning	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Definition of learning systems. Goals and applications of machine learning. Aspects of developing a learning system: training data, concept representation, function approximation. **Inductive Classification:** The concept learning task. Concept learning as search through a hypothesis space. General-to-specific ordering of hypotheses. Finding maximally specific hypotheses. Version spaces and the candidate elimination algorithm. Learning conjunctive concepts. The importance of inductive bias.	**5**
2	**Decision Tree Learning:** Representing concepts as decision trees. Recursive induction of decision trees. **Picking the best splitting attribute:** entropy and information gain.Overfitting, noisy data, and pruning, Linear regression	**5**
3	**Artificial Neural Networks:** Neurons and biological motivation. Perceptrons, Multilayer networks and back propagation. **Bayesian Learning:** Probability theory and Bayes rule. Naive Bayes learning algorithm. Logistic regression	**5**
4	**Support Vector Machine**, Kernel function and Kernel SVM **Instance-Based Learning:** Constructing explicit generalizations versus comparing to past specific examples. k-Nearest Neighbor algorithm. Case-based learning.	**5**

5	**Genetic Algorithm and Evolutionary Algorithms Introduction.** Representing hypothesis, Genetic Operators, Fitness function and selection. Hypothesis space search Genetic Programming	**6**
6	**Clustering and Unsupervised Learning:** Learning from unclassified data. Hierarchical Agglomerative Clustering. k-means partitioned clustering. Expectation maximization (EM) for soft clustering. Semi-supervised learning with EM using labeled and unlabeled data.	**6**
Total lectures		**32**

Suggested Text Book(s):

1. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997, ISBN 0070428077
2. Introduction to Machine Learning Edition 2, by Ethem Alpaydin.

Suggested Reference Book(s):

1. Richard o. Duda, Peter E. Hart, and David G. Stork, “Pattern Classification”, John Wiley Asia, 2006
2. T. Hastie, R. Tibshirani, & J. H. Friedman, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction”, Springer Verlag, 2001.
3. Ian H. Witten & Eibe Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations”, Morgan Kaufmann, 1999.
4. S. M. Weiss & C. A. Kulikowski, “Computer Systems that Learn”, Morgan Kaufman Publishers, San Fancisco, CA, 1991

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc18_cs40/preview
2. Link to topics related to course:
    i. https://in.udacity.com/course/intro-to-machine-learning--ud120-india
    ii. https://www.edx.org/learn/machine-learning
    iii. https://www.datacamp.com/courses/introduction-to-machine-learning-with-r
    iv. https://www.simplilearn.com/big-data-and-analytics/machine-learning-certification-training-course

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Machine Learning)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-4	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
CO-5	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.8**	**2.8**	**2.2**	**2**	**2.2**	**3**	**1**	**3**	

C# and VB.NET

COURSE CODE: 18B1WCI637

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: C/C++ and OOPs

Course Objectives:

1. Knowledge of .NET Framework.
2. Programming in C # and VB.NET in Visual Studio Environment
3. Knowledge of object-oriented programming in the C # and VB.NET languages
4. Programming for windows application development
5. Programming for web application development

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Comprehensive knowledge of .NET Framework, C#.NET and VB.NET	Familiarity
CO-2	Knowledge of C#.NET and VB.NET languages.	Assessment
CO-3	Students will able to develop windows applications using C# and VB.NET	Assessment
CO-4	Students will able to develop web applications using ASP.NET with C# and VB.NET	Assessment
CO-5	Develop a data driven windows and web application	Usage

Course Contents:

Unit	Contents	Lectures Required
1	**.NET Framework**: Introduction to C#.NET, VB.NET and VS.NET, Features of C#.NET,VB.NET and VS.NET, The Common Language Runtime (CLR), Memory Management, Cross Language Integration, metadata and the IL Disassembler. C# and VB.NET basics ,Class, Objects, Inheritance, Polymorphism, Error Handling, Common Type System (CTS), .NET Framework Class Library (FCL), Microsoft Intermediate Language(MSIL), Just In Time(JIT) Compiler, Garbage Collection	7
2	**Windows Programming:** Intro to C# and VB.NET GUI dev in Visual Studio, Windows Forms and built in controls, Delegates and Events, Common Controls, Button, CheckBox, Label, Dialog boxes, TreeView and ListView, Custom Controls etc.	7
3	**ASP.NET using C# and VB.NET**: Introduction of ASP.NET using C# and VB.NET, Concept of Web Applications, ASP.NET Architecture, Page Composition Parts ASP.NET, Page Life Cycle, Page Life Cycle Events, ASP.NET Server Controls, HTML Server Controls, Web Server Controls, List Controls, Validation Controls, User Controls & their uses, Navigation Controls, Login Controls, Custom Controls	11
4	**ADO.NET:** Introduction of SQL, Components of SQL, Basic SQL Commands, Data Binding in ASP.Net, Data Binding Expressions, Data Sources & Controls, Insert, Update, Delete Operations using Data Source Controls, Working with Grid View Control Data bound controls DetailsView control **Security and Deployment:** Security in the .NET framework and Deployment in the .NET	7

Total lectures	**32**

Suggested Text Book(s):

1. .NET Framework Essentials, 3rd Edition by By Hoang Lam, Thuan Thai Publisher: O'Reilly Media
2. Head First C#, 3rd Edition By Andrew Stellman, Jennifer Greene Publisher: O'Reilly Media
3. ASP.NET 4 Unleashed 1st Edition, by Stephen Walther

Suggested Reference Book(s):

1. Pankaj Agrawal – Principal of .Net Framwork
2. Vaya Kogent - .NET Programming Black Book – Wiley
3. VB.NET Black Book by Steven Holzner –Dreamtech
4. VB.NET –Wrox Publication
5. C# programming Black Book by Matt Telles

Other useful resource(s):

1. Link to topics related to course:
   - **i.** https://docs.microsoft.com/en-us/dotnet/
   - **ii.** https://msdn.microsoft.com/en-us/library/ff361664(v=vs.110).aspx
   - **iii.** https://msdn.microsoft.com/en-us/library/aa286485.aspx
   - **iv.** https://blogs.msdn.microsoft.com/dotnet/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (C# and VB.NET)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	21	3	**4.2**
CO-3	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-4	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
CO-5	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.8**	**2.8**	**2.2**	**2**	**2.2**	**3**	**5**	**3**	

Data Structure and Software Design

COURSE CODE: 18B1WCI631

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Data Structures, Compilers, Operating Systems, Computer Networks, Machine Learning and Genetic Algorithms.

Course Objectives:

1. Apply array and stack data structures in mathematical problem solving and implementing various compiler and operating systems mechanisms, respectively.
2. Apply queue and linked list to implement various compiler and operating systems mechanisms.
3. Apply tree and graphs to implement various compiler, computer networks and real time google map applications.
4. Apply priority queue to implement scheduling jobs, sorting huge files and shortest path computation. Apply hashing in implementation of programming languages, file systems, pattern search, and distributed system concepts.
5. Apply data structures in Computer Networks, databases, and image & computer vision.
6. Understanding data structures used in audio/video files, 2D/3D maps, and machine learning & genetics.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn to apply array and stack data structures in mathematical problem solving and implementing various compiler and operating systems mechanisms, respectively.	Assessment
CO-2	To learn to apply queue and linked list to implement various compiler and operating systems mechanisms.	Assessment
CO-3	To learn to apply Apply tree and graphs to implement various compiler, computer networks and real time google map applications.	Assessment
CO-4	To learn to apply priority queue to implement scheduling jobs, sorting huge files and shortest path computation. Apply hashing in implementation of programming languages, file systems, pattern search, and distributed system concepts.	Assessment
CO-5	To learn to apply data structures in Computer Networks, databases, and image & computer vision.	Assessment
CO-6	To learn and understand data structures used in audio/video files, 2D/3D maps, and machine learning & genetics.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Arrays:** finding maximum, minimum, mean, median, average; performing operations – sorting, merging, traversal and retrieval; representing matrices in computer‟s memory for solving many complex mathematical problems and image processing transformations **Stacks:** evaluating arithmetic expressions, storing function arguments and local data as programs are executed, storing local variables used inside a function block in compilers/operating systems, undo mechanisms in text editors, backtracking in	7

	a series of elements, parsing computer languages in compilers, processing function calls and implementing recursive functions, creating space for parameters and local variables in language processing, compiler‟s syntax check for matching braces	
2	**Queues:** interrupts in operating system, how application programs store incoming data, process synchronization in operating system, CPU job scheduling and disk scheduling, **Linked lists:** dynamic memory management, representing polynomials and performing addition, subtraction, multiplication, etc. with polynomials, symbol tables for balancing parenthesis and representing sparse matrix	**5**
3	**Trees:** syntax validation in compilers, implementing sorted dictionary, usage in Internet protocols, storing router tables, quick traversal and searching of directory structures **Graphs:** representing link structure of a website using directed graph, job scheduling problems of CPU, simultaneous execution of jobs problem between set of processors and set of jobs, real time applications of data structures: determination of cities using google maps to find population, finding addresses on maps	**5**
4	**Priority queues:** efficiently schedule jobs (either in the context of a computer operating system or in real life), sorting huge files (which are the most important building block for any Big Data processing algorithm), and efficiently computing shortest paths in graphs. Minimum spanning trees greedy algorithms: Kruskal‟s and Prim‟s **Hashing:** implementating file systems, pattern search, distributed key-value storage, hash functions used in modern distributed systems optimizing storage of services like Dropbox, Google Drive and Yandex Disk!	**5**
5	**Data structure for Computer Networks-** Routing tables, DNS Query, IP Config, Netstat, Address IP, Hashing for MAC tables, protocols **Data structure for popular databases:** Representing indexes with B- trees, buffer trees, quad trees, R-trees, interval trees, hashing etc. **Image & Computer Vision-** Data structure for image representation, pattern recognition, for object recognition – face, house activity analysis, Disjoint sets data structure in dynamic graph connectivity and image processing	**5**
6	**Data structure for video and audio systems:** Video – MPEG-4 Video File (.mp4), MPEG Video File (.mpg), Adobe Flash Video (.flv), Windows Media Video File (.wmv), Uncompressed Audio Formats – Waveform Audio File Format (WAV), Audio Interchange File Format (AIFF), Lossy Compressed Audio Formats – MPEG-1 Audio Layer-3 (MP3), Windows Media Audio (WMA), Lossless Compressed Audio	**5**
	Formats – Free Lossless Audio Codec (FLAC), Apple Lossless Audio Codec (ALAC), Windows Media Audio (WMA). **Data structure for 2D/3D maps:** Vector and Raster data structures, Entity-by-entity data structure (point, line, area), Topological data structures, Tessellations and the tin, Quad Trees, Maps as matrices, Map and attributes **Learned Data structures for machine learning algorithms:** Learned indexes (Cumulative Distribution Functions (CDFs)), Learned Hash-Maps, Learned Bloom filters, Comparison of traditional and learned data structures. Introduction to data structure for genetics	
Total lectures		**32**

Suggested Text Book(s):

1. Horowitz and Sahani: Fundamental of Data Structures in C, 2nd Edn, 2008

Suggested Reference Book(s):

1. Data structures and network algorithms by Robert Endre Tarjan, Society for Industrial and Applied Mathematics (SIAM), ISBN-0-89871-187-8.
2. Computer Networks, 5th edition, Andrew S. Tanenbaum and David J. Wetherall, Pearson Education.
3. Data structure and operating system (Wiley series in computing) by Teodor Rus, John Wiley & Sons Ltd., ISBN-13: 978-0471995173
4. Compiling Techniques by F.R.A. Hopgood, Macdonalds http://www.chilton-computing.org.uk/acl/literature/books/compilingtechniques/contents.htm
5. Compiler construction by William M. Waite and Gerhard Goos, Springer Verlag, ISBN-0-387-90821-8.
6. Data Structures for Databases by Jaochim Hammer and Markus Schneider, https://www.cise.ufl.edu/~mschneid/Research/papers/HS05BoCh.pdf
7. Genetic Algorithms + Data Structures = Evolution Programs by Zbigniew Michalewicz, Springer, ISBN-3-540-60676-9 and ISBN-3-540-58090-5 (2nd edition).
8. Algorithms for graphics and image processing by Theo Pavlidis, Computer Science Press , Inc. Fundamentals of Database Systems"   Elmasri, Navathe, Pearson Education.
9. “Database system concepts” Henry F Korth, Abraham Silberschatz, S. Sudurshan, McGraw-Hill
10. Deo, N., Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, 1974.
11. Digital Image Processing, R.C. Gonzalez and R.E. Woods, 2nd edition, Pearson Prentice Hall, 2008
12. Operating System Concepts by Silberschatz and Galvin, 9th Edition, John Wiley & Sons.

Other useful resource(s):

1. Data structure for 2D/3D maps: http://www.geog.ucsb.edu/~kclarke/AACC/Chapter08.pdf
2. Audio Formats : https://www.makeuseof.com/tag/audio-file-format-right-needs/Video Formats :
3. https://www.lcps.org/cms/lib/VA01000195/Centricity/Domain/1349/Resources/Video_file_formats.pdf

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Data Structure and Software Design)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-2	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-3	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-4	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-5	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-6	3	2	2	3	3	3	3	2	3	2	3	3	**2.7**

Average	**3**	**2.8**	**2.8**	**3**	**2.2**	**3**	**3**	**2**	**3**	**2**	**3**	**3**	

Software Testing Fundamentals Lab

COURSE CODE: 18B1WCI673

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Object Oriented Analysis and Design with UML, Software Engineering, Software Metrics, Basics of Mathematics.

Course Objectives:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
5. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Have an ability to apply software testing knowledge and engineering methods.	Familiarity
CO-2	Have an ability to design and conduct a software test process for a software testing project.	Usage
CO-3	Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.	Assessment
CO-4	Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.	Assessment
CO-5	Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.	Usage
CO-6	Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems.	Usage
CO-7	Have an ability to use software testing methods and modern software testing tools for their testing projects.	Usage

List of Experiments:

S. No	Description	Hours
1	Revision of Java coding using Eclipse	**2**
2	Developing Black box test cases 1: a) Using Boundary Value Analysis	**2**
3	Developing Black box test cases 2:	
	a) Using Equivalent Classes	**2**
4	Developing Black box test cases 3: a) Using DecisionTables	**2**

5	Developing WhiteBox test cases 1: a) Performing Path testing	2
6	Developing WhiteBox test cases 2: a) Performing orthogonal testing	2
7	Developing WhiteBox test cases 3: a) Performing Coverage Analysis	2
8	Mutation testing and developing Mutants	2
9	Regression testing and Developing Regression test Cases	2
10	Performing GUI testing for a designed application	2
11	Performing Load testing for a designed application	2
12	Getting familiar with Profiler and performing CPU, Memory analysis in real time	4
Total Lab Hours		**26**

Suggested Books/Resources:

1. 1. A Practitioner‟s Guide to Software Test Design, Lee Copeland, 2003,
2. The Art of Software Testing, 2nd edition, Glenford Myers, et. el., 2004
3. Software Testing Techniques, 2nd edition, Boris Beizer, 1990
4. How to Break Software: A Practical Guide to Testing, James Whittaker, 2002.
5. Testing Object-Oriented Systems: Models, Patterns, and Tools, Robert V. Binder, 1999.
6. Software Testing and Quality Assurance: Theory and Practice Paperback – 2010

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO 12	Average
CO-1	3	2	3	2	2	3	3	3	2	3	2	2	**2.5**
CO-2	3	3	3	2	3	3	3	3	2	3	2	3	**2.8**
CO-3	3	3	3	2	2	3	3	3	3	3	2	2	**2.7**
CO-4	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-5	3	3	3	2	2	2	3	3	3	3	2	2	**2.6**
CO-6	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-7	3	3	3	3	3	3	2	2	3	3	3	3	**2.8**
Average	**3**	**2.9**	**3**	**2.4**	**2.6**	**2.9**	**2.9**	**2.6**	**2.4**	**3**	**2.4**	**2.6**	

Machine Learning Lab

COURSE CODE: 18B1WCI674

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. To understand the working of Machine learning tools and languages.
2. To learn the implementation of classification techniques for any dataset.
3. To conduct experiments for clustering techniques for any dataset.
4. To discuss different classification and clustering algorithms based on the analysis of results obtained from experimental evaluation.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To implement classification algorithms in python	Usage
CO-2	To implement Clustering algorithms in python	Usage
CO-3	To implement Genetic Algorithms in Python	Usage
CO-4	Top compare different algorithms based on some common factors	Assessment

List of Experiments:

S. No.	Description	Hours
1	Selection of dataset and brief introduction about Python framework for machine learning experiments.	**2**
2	Dataset upload and visualization in Python. • How to upload a dataset in Python • How to retrieve rows and data in the dataset • How to create visualization of data	**2**
3	Decision tree using Entropy and Information Gain	**2**
4	Random forest tree and evaluation of Decision Tree	**2**
5	Linear Regression	**2**
6	Artificial Neural Network • Perceptron • Multi-Layer Neural Network • Back propagation	**4**
7		
8	Naive Bayes Classifier	**2**
9	Logistic regression	**2**
10	Support Vector Machine, Kernel function and Kernel SVM	**2**
11		
12	Genetic Programming	**2**
13		
14	Clustering: k-means	**2**
Total Lab hours		**26**

Approved in Academic Council held on 28 July 2021

Suggested Books/Resources:

1. Tom Mitchell, “Machine Learning”, McGraw Hill, 1997, ISBN 0070428077
2. Sebastian Raschka, “Python Machine Learning”, Packt Publishing Ltd.
3. Andreas C. Müller, Sarah Guido, “Introduction to Machine Learning with Python”, O'Reilly Media, Inc.
4. Sunila Gollapudi, “Practical Machine Learning”, Packt Publishing Ltd
5. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, O'Reilly Media, Inc.
6. Willi Richert, “Building Machine Learning Systems with Python”, Packt Publishing Ltd.
7. Link to topics related to course:
   1. https://www.python-course.eu/machine_learning.php
   2. https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/
   3. https://www.datacamp.com/

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	2	2	3	1	1	1	1	1	2	2	**1.8**
CO-2	3	3	3	3	3	2	1	2	2	1	2	3	**2.3**
CO-3	3	3	2	3	3	2	1	2	2	1	2	3	**2.3**
CO-4	3	3	3	3	3	2	1	2	2	1	2	3	**2.3**
Average	**3**	**2.8**	**2.5**	**2.8**	**3**	**1.8**	**1**	**1.8**	**1.8**	**1**	**2**	**2.8**	

C# and VB.NET Lab

COURSE CODE: 18B1WCI677

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Knowledge of .NET Framework.
2. Programming in C # and VB.NET in Visual Studio Environment
3. Knowledge of object-oriented programming in the C # and VB.NET languages
4. Programming for windows application development
5. Programming for web application development

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Comprehensive knowledge of .NET Framework, C#.NET and VB.NET.	Assessment
CO-2	Knowledge of C#.NET and VB.NET languages.	Assessment
CO-3	Students will able to develop windows applications using C# and VB.NET.	Assessment
CO-4	Students will able to develop web applications using ASP.NET with C# and VB.NET .	Usage
CO-5	Develop a data driven windows and web application.	Usage

List of Experiments:

Experiment	Details	Lab Hours
Experiment 1	Implementation of Class (class, objects, member variable, properties, member functions, constructors)	**2**
Experiment 2	Implementation of Inheritance (Single , Multilevel,)	**2**
Experiment 3	Implementation of Polymorphism (virtual function, overriding, abstract class)	**2**
Experiment 4	Develop GUI Scientific Calculator.	**2**
Experiment 5	Develop a Mini Word application using menu controls.	**2**
Experiment 6	Develop mini browser application using browser control	**2**
Experiment 7	Create simple web form for registration page using standard web controls	**2**
Experiment 8	Apply validation controls on registration page	**2**
Experiment 9	ASP.NET Page Life Cycle, ASP.NET Page Life Cycle Events	**2**
Experiment 10	Create 5 ASP.NET pages (Home, Department, Library, Downloads, Contact us) and apply navigational controls using site map	**2**
Experiment 11	Create Table in SQL Server for registration page created in experiment 7 and apply database connectivity i.e., on click submit button data should be inserted to Table.	**2**
Experiment 12	Create Login Page with database connectivity	**2**

Approved in Academic Council held on 28 July 2021

Experiment 13	Create asp.net page for listing all the registered users in Grid View	**2**
Experiment 14	Apply functionality of Searching, Creation, Updating and Deletion of registered users.	**2**
TOTAL Lab hours		**28**

Suggested Books/Resources:

1. NET Framework Essentials, 3rd Edition by By Hoang Lam, Thuan Thai Publisher: O'Reilly Media
2. Head First C#, 3rd Edition By Andrew Stellman, Jennifer Greene Publisher: O'Reilly Media
3. ASP.NET 4 Unleashed 1st Edition, by Stephen Walther
4. Pankaj Agrawal – Principal of .Net Framwork
5. Vaya Kogent - .NET Programming Black Book – Wiley
6. VB.NET Black Book by Steven Holzner –Dreamtech
7. VB.NET –Wrox Publication
8. C# programming Black Book by Matt Telles
9. Link to topics related to course:
   a. https://docs.microsoft.com/en-us/dotnet/
   b. https://msdn.microsoft.com/en-us/library/ff361664(v=vs.110).aspx
   c. https://msdn.microsoft.com/en-us/library/aa286485.aspx
   d. https://blogs.msdn.microsoft.com/dotnet/

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-2	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-3	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-4	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-5	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
Average	**3**	**3**	**3**	**3**	**2**	**3**	**3**	**2**	**3**	**2**	**3**	**3**	

Data Structure and Software Design Lab

COURSE CODE: 18B1WCI671

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Apply array and stack data structures in mathematical problem solving and implementing various compiler and operating systems mechanisms, respectively.
2. Apply queue and linked list to implement various compiler and operating systems mechanisms.
3. Apply tree and graphs to implement various compiler, computer networks and real time google map applications.
4. Apply priority queue to implement scheduling jobs, sorting huge files and shortest path computation. Apply hashing in implementation of programming languages, file systems, pattern search, and distributed system concepts.
5. Apply data structures in Computer Networks, databases, and image & computer vision.
6. Understanding data structures used in audio/video files, 2D/3D maps, and machine learning & genetics.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn to apply array and stack data structures in mathematical problem solving and implementing various compiler and operating systems mechanisms, respectively.	Assessment
CO-2	To learn to apply queue and linked list to implement various compiler and operating systems mechanisms.	Assessment
CO-3	To learn to apply Apply tree and graphs to implement various compiler, computer networks and real time google map applications.	Assessment
CO-4	To learn to apply priority queue to implement scheduling jobs, sorting huge files and shortest path computation. Apply hashing in implementation of programming languages, file systems, pattern search, and distributed system concepts.	Assessment
CO-5	To learn to apply data structures in Computer Networks, databases, and image & computer vision.	Assessment
CO-6	To learn and understand data structures used in audio/video files, 2D/3D maps, and machine learning & genetics.	Usage

List of Experiments:

S. No.	Description	Hours
1	Using arrays to (a) find maximum, minimum, mean, median, average (b) performing operations – sorting, merging, traversal and retrieval (c) complex mathematical problems and image processing transformations	3
2	Using stack to (a) evaluating arithmetic expressions (b) storing function arguments and local data as programs are executed	4

	(c) storing local variables used inside a function block in compilers/operating systems (d) undo mechanisms in text editors (e) backtracking in a series of elements (f) parsing computer languages in compilers (g) processing function calls and implementing recursive functions (h) creating space for parameters and local variables in language processing, compiler‟s syntax check for matching braces	
3	Using queue to (a) handle interrupts in operating system (b) store incoming data in application programs (c) process synchronization in operating system (d) CPU job scheduling and disk scheduling	**3**
4	Using linked list to (a) dynamic memory management (b) representing polynomials and performing addition, subtraction, multiplication, etc. with polynomials (c) symbol tables for balancing parenthesis and representing sparse matrix	**3**
5	Using trees to (a) syntax validation in compilers (b) implementing sorted dictionary (c) usage in Internet protocols (d) storing router tables (e) quick traversal and searching of directory structures	**3**
6	Using graphs to (a) representing link structure of a website using directed graph (b) job scheduling problems of CPU (c) simultaneous execution of jobs problem between set of processors and set of jobs (d) real time applications of data structures: determination of cities using google maps to find population, finding addresses on maps	**3**
7	Using priority queue to (a) efficiently schedule jobs (b) sorting huge files (c) efficiently computing shortest paths in graphs (d) Minimum spanning trees greedy algorithms: Kruskal‟s and Prim‟s	**3**
8	Using hashing to (a) implement file systems and pattern search (b) distributed key-value storage and hash functions used in modern distributed systems optimizing storage of services like Dropbox, Google Drive and Yandex Disk!	**2**
9	Implementing data structures for (a) Computer Network protocols and routing tables. (b) Indexes used in databases (c) Image and computer vision	**2**
10	Implementing data structures for (a) Audio and video files (b) 2D/3D maps (c) Learned data structures for Machine Learning algorithms	**2**
Total Lab Hours		**28**

Suggested Books/Resources:

1. Horowitz and Sahani: Fundamental of Data Structures in C, 2nd Edn, 2008
2. Data structures and network algorithms by Robert Endre Tarjan, Society for Industrial and Applied Mathematics (SIAM), ISBN-0-89871-187-8.
3. Computer Networks, 5th edition, Andrew S. Tanenbaum and David J. Wetherall, Pearson Education.
4. Data structure and operating system (Wiley series in computing) by Teodor Rus, John Wiley & Sons Ltd., ISBN-13: 978-0471995173
5. Compiling Techniques by F.R.A. Hopgood, Macdonalds
6. http://www.chilton-computing.org.uk/acl/literature/books/compilingtechniques/contents.htm

7. Compiler construction by William M. Waite and Gerhard Goos, Springer Verlag, ISBN-0-387-90821-8.
8. Data Structures for Databases by Jaochim Hammer and Markus Schneider, https://www.cise.ufl.edu/~mschneid/Research/papers/HS05BoCh.pdf
9. Genetic Algorithms + Data Structures = Evolution Programs by Zbigniew Michalewicz, Springer, ISBN-3-540-60676-9 and ISBN-3-540-58090-5 (2nd edition).
10. Algorithms for graphics and image processing by Theo Pavlidis, Computer Science Press , Inc. Fundamentals of Database Systems" Elmasri, Navathe, Pearson Education.
11. “Database system concepts” Henry F Korth, Abraham Silberschatz, S. Sudurshan, McGraw-Hill
12. Deo, N., Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, 1974.
13. Digital Image Processing, R.C. Gonzalez and R.E. Woods, 2nd edition, Pearson Prentice Hall, 2008
14. Operating System Concepts by Silberschatz and Galvin, 9th Edition, John Wiley & Sons.
15. Data structure for 2D/3D maps: http://www.geog.ucsb.edu/~kclarke/AACC/Chapter08.pdf
16. Audio Formats : https://www.makeuseof.com/tag/audio-file-format-right-needs/Video Formats :
17. https://www.lcps.org/cms/lib/VA01000195/Centricity/Domain/1349/Resources/Video_file_formats.pdf

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PO12	Average
CO-1	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-2	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-3	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-4	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-5	3	3	3	3	2	3	3	2	3	2	3	3	**2.8**
CO-6	3	2	2	3	3	3	3	2	3	2	3	3	**2.7**
Average	**3**	**2.8**	**2.8**	**3**	**2.2**	**3**	**3**	**2**	**3**	**2**	**3**	**3**	

DETAILED COURSE DESCRIPTIONS

ELECTIVE - III

Pattern Recognition

COURSE CODE: 18B1WCI638
COURSE CREDIT: 2
CORE/ELECTIVE: ELECTIVE
L-T-P: 2-0-0

Pre-requisites: Students who have just finished a first course in programming. Knowledge of writing programs in any programming language is expected. No prior experience with data structuers is required.

Course Objectives:

1. This course will introduce the fundamentals of statistical pattern recognition.
2. Generative methods such as those based on Bayes decision theory and related techniques of parameter estimation and density estimation.
3. Discussion of discriminative methods such as nearest-neighbor classification and support vector machines.
4. Applications such as information retrieval, data mining, document image analysis and recognition, computational linguistics, forensics, biometrics and bioinformatics with pattern recognition.

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	Understanding of the fundamentals of statistical pattern recognition.	Familiarity
CO-2	Generative methods such as those based on Bayes decision theory and related techniques of parameter estimation and density estimation.	Assessment
CO-3	Discussion of discriminative methods such as nearest-neighbor classification and support vector machines.	Assessment
CO-4	Clustering of data and related algorithms are to be learned.	Assessment
CO-5	Clustering in large databases and related algorithms are to be learned.	Assessment
CO-6	Combinations of Classifiers are to be understood and learned with applications.	Usage
CO-7	Applications such as information retrieval, data mining, document image analysis and recognition, computational linguistics, forensics, biometrics and bioinformatics with pattern recognition.	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	**Introduction –** Definitions, data sets for Pattern Recognition, Different Paradigms of Pattern Recognition, Representations of Patterns and Classes, Metric and non-metric proximity measures	6
2	**Feature extraction**, Different approaches to Feature Selection	3
3	**Nearest Neighbour Classifier** and variants, Efficient algorithms for nearest neighbor classification	3
4	**Different Approaches** to Prototype Selection, Bayes Classifier, Decision Trees, Linear Discriminant Function, Support Vector Machines	10

5	**Clustering**, Clustering Large datasets, Combination of Classifiers	7
6	**Applications** – Document Recognition	1
Total Lectures		**32**

Suggested Text Book(s):

1. Devi V.S.; Murty, M.N. (2011) Pattern Recognition: An Introduction, Universities Press, Hyderabad.
2. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley, 2000.

Suggested Reference Book(s):

1. Theodoridis, Koutroumbas: Pattern Recognition, 2nd ed., Elsevier, Amsterdam, 2003
2. C.M. Bishop: Pattern Recognition and Machine Learning. Springer Verlag, Singapore, 2006.
3. C.M. Bishop: Neural Networks for Pattern Recognition. Clarendon Press, Oxford, 1996.
4. R. Schalkoff: Pattern Recognition. Statistical, Structural, and Neural Approaches. John Wiley & Sons, Inc., 1992.

Other useful resource(s):

1. Link to NPTEL course contents: http://nptel.iitm.ac.in
2. Link to topics related to course:
   i. https://www5.cs.fau.de/
   ii. https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/intelligent-systems/pattern-recognition-bioinformatics/pattern-recognition-laboratory/
   iii. https://www.dei.unipd.it/node/370

Evaluation Scheme:

S. No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Pattern Recognition)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	2	3	2	2	3	2	3	2	2	3	3	**2.5**
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	**2.6**
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	**2.5**
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	**2.4**
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	**2.6**
CO-7	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
Average	**2.6**	**2.1**	**2.6**	**2.1**	**2.4**	**2.6**	**2.6**	**2.6**	**2.1**	**2.6**	**2.9**	**2.7**	

Data Mining and Data Warehousing

COURSE CODE: 18B1WCI635

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Data Structures, Compilers, Operating Systems, Computer Networks, Machine Learning and Genetic Algorithms.

Course Objective:

1. To describe the concept of Data warehouse & its attributes
2. To study different data warehouse models, architectures and implementation
3. To understand the basic concept of data mining and its functionality
4. To understand the concept of classification techniques and its implementation
5. To understand the concept of association rules, different techniques and implementation details
6. To understand the concept of cluster analysis, anomaly detection and its usage and implementation details

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To describe the concept of Data warehouse & its attributes	Assessment
CO-2	To study different data warehouse models, architectures and implementation	Assessment
CO-3	To understand the basic concept of data mining and its functionality	Assessment
CO-4	To understand the concept of classification techniques and its implementation	Assessment
CO-5	To understand the concept of association rules, different techniques and implementation details	Assessment
CO-6	To understand the concept of cluster analysis, anomaly detection and its usage and implementation details	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Concepts of Data Warehouse and Data Mining including its functionalities, stages of Knowledge discovery in database(KDD) , Setting up a KDD environment, Issues in Data Warehouse and Data Mining, Application of Data Warehouse and Data Mining	**5**
2	**Architecture:** DBMS vs. Data Warehouse, Data marts, Metadata, Multidimensional data model, Data Cubes, Schemas for Multidimensional Database: Stars, Snowflakes and Fact Constellations, Data Warehouse Architecture, Distributed and Virtual Data Warehouse, Data Warehouse Manager, OLTP, OLAP, MOLAP, HOLAP, types of OLAP, servers	**6**
3	**Introduction:** Data Mining, Motivation, Challenges, Origins of Data Mining, Data Mining Tasks, Data: Types of Data, Data Quality, Data Pre-processing, Measures of Similarity and Dissimilarity, Exploring Data: Iris Data Set, Summary Statistics,	**3**

	Visualization, OLAP and Multi dimensional Data Analysis	
4	**Classification:** Basic Concepts and Preliminaries, Approach to Solving a Classification Problem, Decision Tree Induction, Model Over fitting, Evaluating Performance of Classifier **Alternative Techniques**: Rule-Based Classifier, Nearest- Neighbour Classifiers, Artificial Neural Network (ANN), Support Vector Machine (SVM), Ensemble Methods, Class Imbalance Problem, Multiclass Problem	**6**
5	**Association Analysis:** Basic Concepts and Problem Definition, Frequent Itemset Generation, Rule Generation, Representation of Frequent Itemsets, FP-Growth Algorithm, Evaluation of Association Patterns, Handling Categorical Attributes , Handling Continuous Attributes, Handling a Concept Hierarchy, Sequential Patterns, Subgraph Patterns	**6**
6	**Cluster Analysis:** Basic Concepts, Characteristics of Data, Clusters, Partitional Clustering, Agglomerative Hierarchical Clustering, Prototype-Based Clustering, Density-Based Clustering, Graph-Based Clustering, Cluster Evaluation **Anomaly Detection:** Preliminaries, Statistical Approaches, Proximity-Based Outlier Detection, Density-Based Outlier Detection, Clustering-Based Techniques	**6**
Total lectures		**32**

Suggested Text Book(s):

1. Data Mining Concepts and Techniques J. Han and M. Kamber Morgan Kaufmann, 2006, ISBN 1-55860-901-6

2. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education (Addison Wesley), 0-321-32136-7

Suggested Reference Book(s):

1. Mining Massive data sets Anand Rajaram, Jure Leskovec and Jeff Ullman Cambridge University Press
2. https://onlinecourses.nptel.ac.in/noc18_cs14/preview
3. https://www.coursera.org/specializations/data-mining.
4. https://www.futurelearn.com/courses/data-mining-with-weka

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Data Mining and Data Warehousing)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO 1	3	2	1	2	2	2	2	2	2	1	1	2	**1.8**

CO 2	3	3	3	3	3	3	2	3	3	1	1	3	**2.6**
CO 3	3	3	3	3	2	2	1	2	3	2	3	3	**2.5**
CO 4	3	1	2	1	2	1	1	2	1	2	2	2	**1.7**
CO 5	3	3	3	3	2	3	1	2	3	2	3	3	**2.6**
CO 6	3	2	1	2	2	1	2	2	2	1	1	2	**1.8**
Average	**3**	**2.3**	**2.2**	**2.3**	**2.2**	**2**	**1.5**	**2.2**	**2.3**	**1.5**	**1.8**	**2.5**	

Parallel and Distributed Algorithms

COURSE CODE: 18B1WCI632

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: You should be comfortable programming in C and /or Java in particular. No prior knowledge of parallel computing is required. Good knowledge of undergraduate level algorithms, data structures, operating system and computer architecture.

Course Objectives:

1. To acquaint students with the basic concepts of parallel and distributed computing.
2. To learn general principles of parallel and distributed algorithms
3. To analyse their time complexity.
4. To acquaint students with various parallel and distributed approaches of problem solving.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To reason about ways to parallelize a problem and be able to evaluate a parallel platform for a given problem.	Familiarity
CO-2	To understand and explore the concepts with programming with MPI and Map Reduce/Hadoop.	Assessment
CO-3	To demonstrate the general concepts on Cloud computing, grid computing, and peer-to-peer systems.	Usage
CO-4	To become familiar with evaluation of online social networks and their potential.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction: The power and potential of parallelism** The power and potential of parallelism, purpose of using parallelism, different parallel architecture, Reasoning about performance of parallel programs.	6
2	**Data, Task Parallelism and Java Multithreading** Introduction of data and task parallelism, Independent parallelism, Introduction to Java multithreading, Fork-join parallelism, Analyze fork and join parallelism, parallel prefix, parallel pack	6
3	**Mutual exclusion, Deadlocks and Parallel Computational Models** Concurrency, STM, Mutual exclusion, locks, Deadlocks, race condition, Read/write locks, condition variables, Flynn‟s Taxonomy, PRAM, EREW, CREW, ERCW, CRCW, Simulating CRCW, CREW and EREW, PRAM algorithms. Parallel Programming Models, PVM, MPI Paradigms	6
4	**Parallel Algorithms and Programming Languages** Parallel Programming Language, Brent‟s Theorem, Simple parallel programs in MPI environments, Parallel algorithms on network, Addition of Matrices, Multiplication of Matrices., Parallel quick sort, Synchronizing shared data structure, Shared memory	6
5	**Distributed System Model and Cases** Distributed system models, Inter process communication, Message passing, Message passing algorithm, Distributed synchronization, Consistency,	8

	replication, Cluster computing, MapReduce, Distributed storage, Wide area computing, Distributed hash table, Peer-to-peer systems. Cases a) Parallel computing algorithms and representative programming models, b) Convergence of parallel, distributed and cloud computing, c) Cluster Computing, its performance model and system evolution.	
Total lectures		**32**

Suggested Text Book(s):

1. “A. Grama, A. Gupta, G. Karypis and V. Kumar. Introduction to Parallel Computing (2nd edition), Addison Wesley (2002), ISBN 0-201-64865-2.
2. H. El-Rewini and T.G. Lewis. Distributed and Parallel Computing, Manning (1997), ISBN 0-13-795592-8.
3. I. Foster. Designing and Building Parallel Programs, Addison Wesley (1995), ISBN 0-201-57594-9.

Suggested Reference Book(s):

1. Kai Hwang and Zhiwei Xu. Scalable Parallel Computing, McGraw Hill (1998), ISBN 0-07-031798-4.
2. Michael J. Quinn. Parallel Programming in C with MPI and OpenMP, McGraw Hill (2003), ISBN 0-07-282256-2.
3. Barry Wilkinson and Michael Allen. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition), Prentice Hall PTR (2005), ISBN 0-13-140563-2

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106102114/
2. Link to topics related to course: https://nptel.ac.in/courses/106106107/
   i. https://nptel.ac.in/courses/106106112/
   ii. https://nptel.ac.in/courses/106106112/2
   iii. https://nptel.ac.in/courses/106106112/3
   iv. https://nptel.ac.in/courses/106106112/3

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Parallel and Distributed Algorithms)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	2	2	2	2	1	1	1	2	1	1	2	**1.6**

CO-2	2	3	3	3	3	1	1	1	2	2	1	1	**1.9**
CO-3	2	3	3	3	3	1	1	1	2	2	1	1	**1.9**
CO-4	2	3	3	3	2	1	1	1	1	3	2	2	**2**
Average	**2**	**2.8**	**2.8**	**2.8**	**2.5**	**1**	**1**	**1**	**1.8**	**2**	**1.3**	**1.5**	

Digital Image Processing

COURSE CODE: 18B1WCI636

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Linear algebra, Matrices, Matrix Operations, Determinants, Systems of Linear Equations, Eigen values, Eigenvectors, Statistics and probability, Programming experience, preferably in matlab, and/or C/C++/C#/Python/Java

Course Objectives:

1. Introduction to various image processing techniques.
2. Learning the basics of Image fundamentals, describing the main characteristics of digital images and how they are represented.
3. Learning of mathematical transforms such as such as Fourier, Cosine transforms, Singular value decomposition.
4. Understanding the concepts of 2D Wavelet transform, image enhancement techniques, Image restoration and denoising, segmentation.
5. Discussing and understanding the concepts of lossy and lossless data compression algorithms, binary and color image processing.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To learn the basic concepts and terminology in digital Image Processing.	Familiarity
CO-2	To learn about image transformation techniques and issues related to image transformation.	Assessment
CO-3	To learn the basic techniques for Image Compression	Assessment
CO-4	To learn about enhancing images through techniques like filtering and equalization	Assessment
CO-5	To learn about image restoration, segmentation and denoising.	Assessment
CO-6	To implement graph theory in vector space models and colouring methodologies for images in MATLAB	Usage
CO-7	To understand image processing techniques" case studies	Familiarity

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Digital Image Processing** Introduction to images and its processing, Components of image processing systems, image representations, Image file formats, Applications of digital image processing, image sampling and quantization, Image Analysis, Intensity transformations, contrast stretching, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.	5
2	**Image Transformation Techniques** Need for transform, Fourier, Cosine transforms, Haar, KL Transform, Singular value decomposition, 2D Wavelet transform,	8

	Different properties of image transform techniques.	
3	**Image Compression Basics** Concept of image compression, lossless techniques (Huffman Coding, Arithmetic and Lempel-Ziv Coding, Other Coding Techniques) and lossy compression techniques (Transform Coding & K-L Transforms, Discrete Cosine Transforms, and BTC), Multi-Resolution Analysis, and Still Image Compression Standards (JBIG and JPEG),	**8**
4	**Image Enhancement** Enhancement in spatial and transform domain, histogram equalization Directional Smoothing, Median, Geometric mean, Harmonic mean, Contraharmonic mean filters, Homomorphicfiltering,Color image enhancement.	**5**
5	**Image Restoration and Denoising** Image degradation, Type of image blur, Classification of image restoration techniques, ,image restoration model, Linear and non linear restoration techniques, Image denoising, Median filtering	**5**
6	**Image Segmentation** Classification of image segmentation techniques, Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Region- based segmentation, Watershed algorithm, Use of motion in segmentation	**6**
Total lectures		**32**

Suggested Text Book(s):

1. Digital Image Processing, R.C. Gonzalez and R.E. Woods, 2nd edition, Pearson Prentice Hall, 2008.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989.

Suggested Reference Book(s):

1. Digital Image processing, S Jayaraman, TMH, 2012
2. William K. Pratt, *Digital Image Processing*, 3rd Edition, John Wiley, 2001.

Other useful resource(s):

1. Link to NPTEL course contents:https://onlinecourses.nptel.ac.in/noc18_ee40/preview
2. Link to topics related to course:
   i. https://nptel.ac.in/courses/117105079/1
   ii. https://nptel.ac.in/courses/117105079/6
   iii. https://nptel.ac.in/courses/117105079/3
   iv. https://nptel.ac.in/courses/117105079/10
   v. https://nptel.ac.in/courses/117105079/12

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus

4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Digital Image Processing)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	1	1	3	2	3	3	3	3	2	**2.5**
CO-2	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-3	3	3	3	3	3	3	3	3	3	3	3	3	**3**
CO-4	3	3	2	3	3	3	2	1	1	1	3	3	**2.3**
CO-5	3	3	3	1	2	3	2	3	3	1	3	2	**2.4**
CO-6	2	2	2	3	3	3	2	3	3	1	1	3	**2.3**
CO-7	3	3	3	3	3	3	3	3	3	1	3	3	**2.8**
Average	**2.7**	**2.7**	**2.6**	**2.4**	**2.6**	**3**	**2.3**	**2.7**	**2.7**	**1.9**	**2.4**	**2.7**	

Pattern Recognition Lab

COURSE CODE: 18B1WCI678

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisites: None

Course Objectives:

1. Learning about related tools as IU Box, Labs, SPSS, PsycINFO, Web of Science
2. Learning and deployment of Statistics method
3. Learning and deployment of Pattern recognition methodology
4. Learning and deployment of Feature search
5. Learning and deployment about Computational models used in Pattern Recognition

Course Outcomes:

S.No.	**Course outcomes**	**Level of Attainment**
CO-1	Understanding about related tools as IU Box, Labs, SPSS, PsycINFO, Web of Science	Familiarity
CO-2	Explain and compare a variety of pattern classification, structural pattern recognition, and pattern classifier combination techniques.	Assessment
CO-3	Summarize, analyze, and relate research in the pattern recognition area verbally and in writing.	Assessment
CO-4	Apply performance evaluation methods for pattern recognition, and critique comparisons of techniques made in the research literature.	Assessment
CO-5	Apply pattern recognition techniques to real-world problems such as document analysis and recognition.	Assessment
CO-6	Apply Clustering in databases or large databases.	Usage
CO-7	Implement simple pattern classifiers, classifier combinations, and structural pattern recognizers.	Familiarity

List of Experiments:

S.No	**Description**	**Hours**
1	LAB: Introduction to computer resources (IU Box, Labs, SPSS, PsycINFO, Web of Science)	4
2	Statistics I. Data input/output, summary tables, charting	2
3	Statistics 2. T-tests, ANOVA, regression	2
4	Statistics 3–Factorial ANOVAs	2
5	Statistics 4. Repeated measures ANOVAs	2
6	Feature Representation	2
7	Mean and Covariance	2
8	Linear Perceptron Learning	2

Approved in Academic Council held on 28 July 2021

9	Generation of Random Variables	2
10	Bayesian Classification	2
11	MLE: Learning the classifier from data	2
12	Data Clustering: K-Means, MST-based	4
Total Lab hours		**28**

Suggested Books/Resources:

1. Devi V.S.; Murty, M.N. (2011) Pattern Recognition: An Introduction, Universities Press, Hyderabad.
2. R. O. Duda, P. E. Hart and D. G. Stork, Pattern Classification, Wiley, 2000.
3. https://cse20-iiith.vlabs.ac.in/
4. https://www.mathworks.com/discovery/pattern-recognition.html
5. http://nptel.iitm.ac.in
6. Link to topics related to course:
   - https://www5.cs.fau.de/
   - https://www.tudelft.nl/ewi/over-de-faculteit/afdelingen/intelligent-systems/pattern-recognition-bioinformatics/pattern-recognition-laboratory/
   - https://www.dei.unipd.it/node/370

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	3	2	2	3	2	3	2	2	3	3	**2.5**
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	**2.6**
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	**2.5**
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	**2.4**
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	**2.6**
CO-7	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
Average	**2.6**	**2.1**	**2.6**	**2.1**	**2.4**	**2.6**	**2.6**	**2.6**	**2.1**	**2.6**	**2.9**	**2.7**	

Data Warehouse and Data Mining Lab

COURSE CODE: 18B1WCI675

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: Introduction to Programming

Course Objectives:

1. Practical exposure on implementation of well known data mining tasks.
2. Exposure to real life data sets for analysis and prediction.
3. Learning performance evaluation of data mining algorithms in a supervised and an unsupervised setting.
4. Handling a small data mining project for a given practical domain.
5. Develop and apply enthusiasm for learning machine learning tools and techniques.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Practical exposure on implementation of well known data mining tasks.	Familiarity
CO-2	Exposure to real life data sets for analysis and prediction.	Technical skills
CO-3	Learning performance evaluation of data mining algorithms in a supervised and an unsupervised setting.	Computational skills
CO-4	Handling a small data mining project for a given practical domain.	Computational
CO-5	Develop and apply machine learning tools and techniques.	Assessment

List of Experiments:

S. No.	Description	Hours
1	Introduction to Rapid Miner Studio, Weka and R	**2**
2	Creation of a Data Warehouse in Rapid Miner	**2**
3	Apriori Algorithm	**4**
4	FP-Growth Algorithm	**4**
5	K-means Clustering, K-Means ++ Clustering	**4**
6	Hierarchical clustering algorithm	**4**
7	Bayesian Classification	**2**
8	Decision Tree	**2**
9	Feature Reduction using PCA	**2**
10	Model Examination	**2**
Total Lab hours		**28**

Suggested Books/Resources:

1. Introduction to Data Mining Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education (Addison Wesley), 0-321-32136-7, 2006
2. Mininig Massive data sets Anand Rajaram, Jure Leskovec and Jeff Ullman Cambridge University Press
3. Data Mining Concepts and Techniques J. Han and M. Kamber Morgan Kaufmann, 2006, ISBN 1-55860-901-6

4. An Introduction to Information Retrieval, 2008 Cambridge UP.
5. https://onlinecourses.nptel.ac.in/noc18_cs14/preview
6. https://www.coursera.org/specializations/data-mining
7. https://www.futurelearn.com/courses/data-mining-with-weka
8. https://docs.rapidminer.com/
9. https://archive.ics.uci.edu/ml/datasets.html
10. IRIS dataset: http://archive.ics.uci.edu/ml/datasets/Iris
11. Mushroom dataset: http://archive.ics.uci.edu/ml/datasets/Mushroom
12. Breast cancer dataset: http://archive.ics.uci.edu/ml/datasets/Breast+Cancer
13. Car evaluation dataset: http://archive.ics.uci.edu/ml/datasets/Car+Evaluatio

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	1	2	2	2	2	2	2	1	1	2	**1.9**
CO-2	3	3	3	3	3	3	2	3	3	1	1	3	**2.6**
CO-3	3	3	3	3	2	3	1	2	3	2	3	3	**2.6**
CO-4	3	1	2	1	2	1	1	2	1	1	1	2	**1.5**
CO-5	3	3	3	3	2	3	1	2	3	2	3	3	**2.6**
Average	**3**	**2.6**	**2.4**	**2.4**	**2.2**	**2.4**	**1.4**	**2.2**	**2.4**	**1.4**	**1.8**	**2.6**	

Parallel and Distributed Algorithms Lab

COURSE CODE: 18B1WCI672

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. To acquaint students with the basic concepts of parallel and distributed computing.
2. To learn general principles of parallel and distributed algorithms
3. To analyse their time complexity.
4. To acquaint students with various parallel and distributed approaches of problem solving.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	To reason about ways to parallelize a problem and be able to evaluate a parallel platform for a given problem.	Familiarity
CO-2	To understand and explore the concepts with programming with MPI and MapReduce/Hadoop.	Assessment
CO-3	To demonstrate the general concepts on Cloud computing, grid computing, and peer-to-peer systems.	Usage
CO-4	To become familiar with evaluation of online social networks and their potential.	Usage

List of Experiments:

S. No.	Description	Lab Hours
1	Design, develop a program to implement task parallelism with Java multithreading.	**4**
2	Design, develop a program to implement Fork-join parallelism and Analyse fork and join parallelism, parallel prefix, parallel pack.	**4**
3	Design, develop a program for PRAM algorithms.	**4**
4	Design, develop a program to implement Parallel Programming Models, PVM, MPI Paradigms.	**4**
5	Design, develop a program to implement Brent"s Theorem.	**4**
6	Design, develop a program to solve base on MapReduce.	**8**
Total Lab Hours		**28**

Minor Project(s) – (Only for 2 credit lab)

Create a solution for a Complex Engineering Problem by using Apache Hadoop Map/Reduce. Solution should be distributed in nature or use a distributed programming paradigm in its solution domain.

Suggested Books/Resources:

1. "A. Grama, A. Gupta, G. Karypis and V. Kumar. Introduction to Parallel Computing (2nd edition), Addison Wesley (2002), ISBN 0-201-64865-2.
2. H. El-Rewini and T.G. Lewis. Distributed and Parallel Computing, Manning (1997), ISBN 0-13-795592-8.

3. Foster. Designing and Building Parallel Programs, Addison Wesley (1995), ISBN0-201-57594-9.
4. Kai Hwang and Zhiwei Xu. Scalable Parallel Computing, McGraw Hill (1998), ISBN0-07-031798-4.
5. Michael J. Quinn. Parallel Programming in C with MPI and OpenMP, McGraw Hill (2003), ISBN 0-07-282256-2.
6. Barry Wilkinson and Michael Allen. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers (2nd Edition), Prentice Hall PTR (2005), ISBN 0-13-140563-2
7. Link to NPTEL course contents: https://nptel.ac.in/courses/106102114/
8. Link to topics related to course: https://nptel.ac.in/courses/106106107/
   i. https://nptel.ac.in/courses/106106112/
   ii. https://nptel.ac.in/courses/106106112/2
   iii. https://nptel.ac.in/courses/106106112/3
   iv. https://nptel.ac.in/courses/106106112/3

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	Average
CO1	3	3	3	3	2	2	1	1	1	1	1	1	**1.9**
CO2	3	3	3	3	3	1	1	1	1	1	1	3	**2**
CO3	3	3	2	3	2	3	2	1	1	1	2	1	**2**
CO4	3	3	3	2	3	2	1	1	1	1	1	1	**1.8**
CO5	2	2	3	3	3	3	1	1	1	1	1	1	**1.8**
CO6	2	3	3	3	2	2	2	2	2	2	2	2	**2.3**
Average	**2.7**	**2.8**	**2.8**	**2.8**	**2.5**	**2.2**	**1.3**	**1.2**	**1.2**	**1.2**	**1.3**	**1.6**	

Digital Image Processing Lab

COURSE CODE: 18B1WCI676

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Introduction to various image processing techniques.
2. Implementing the basics of Image fundamentals, describing the main characteristics of digital images and how they are represented.
3. Learning and implementing mathematical transforms such as such as Fourier, Cosine transforms Singular value decomposition.
4. Implementing the concepts of 2D Wavelet transform, image enhancement techniques, Image restoration and denoising, segmentation.
5. Discussing and implementing the concepts of lossy and lossless data compression algorithms, binary and color image processing.

Course Outcomes :

S. No.	**Course Outcomes**	**Level of Attainment**
CO-1	To learn the basic concepts and terminology in digital Image Processing.	Familiarity
CO-2	To learn about image transformation techniques and issues related to image transformation.	Assessment
CO-3	To learn the basic techniques for Image Compression	Assessment
CO-4	To learn about enhancing images through techniques like filtering and equalization	Assessment
CO-5	To learn about image restoration, segmentation and denoising.	Assessment
CO-6	To implement graph theory in vector space models and colouring methodologies for images in MATLAB	Usage
CO-7	To understand image processing techniques‟ case studies	Familiarity

List of Experiments:

S. No	**Description**	**Hours**
1	Implementing images colorizations	**1**
2	Implement various techniques for quantization	**1**
3	Implement filtering techniques	**1**
4	Transform images using Laplacian	**2**
5	Decompose images using cosine, discrete transformation	**1**
6	Implementing lossless compression techniques	**2**
7	Implementing Lossy compression methods	**1**
8	Implementing image enhancement using filtering	**2**
9	Implementing image enhancement using spatial domains	**1**
10	Implementing image enhancement using equalization techniques	**2**
11	Implementing enhancements using mean, median, geometric mean, harmonic mean	**2**
12	Implementing Contraharmonic mean filters, Homomorphic filtering	**2**

13	Implementing image restoration techniques	**1**
14	Implementing denoising techniques	**1**
15	Implementing Boundary detection based techniques	**2**
16	Implementing Edge detection based techniques	**2**
17	Implement Hough transform, Thresholding, Iterative thresholding	**2**
18	Implement Multivariable thresholding	**2**
Total Lab hours		**28**

Minor Project(s) – (Only for 2 credit lab)

1. Image Enhancement using Filtering Techniques.
2. Image Reconstruction using Wavelets transform.
3. Dehazing images using HE, Fattal method
4. Underwater Image enhancement using Wavelets and equalization
5. Underwater Panoramic Image enhancement using mosaicking techniques

Suggested Books/Resources:

1. Digital Image Processing, R.C. Gonzalez and R.E. Woods, 2nd edition, Pearson Prentice Hall, 2008.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall, 1989.
3. Digital Image processing, S Jayaraman, TMH, 2012
4. William K. Pratt, *Digital Image Processing*, 3rd Edition, John Wiley, 2001.
5. Link to topics related to course:
   a. https://nptel.ac.in/courses/117105079/1
   b. https://nptel.ac.in/courses/117105079/6
   c. https://nptel.ac.in/courses/117105079/3
   d. https://nptel.ac.in/courses/117105079/10
   e. https://nptel.ac.in/courses/117105079/12
   f. https://nptel.ac.in/courses/117105079/15
   g. https://nptel.ac.in/courses/117105079/21
   h. https://nptel.ac.in/courses/117105079/29

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	1	2	2	2	3	3	3	3	2	**2.5**
CO-2	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-3	3	3	3	3	3	3	3	3	3	3	3	3	**3**
CO-4	3	3	2	3	3	3	2	1	1	3	3	3	**2.5**
CO-5	3	3	3	1	2	3	2	3	3	3	3	2	**2.6**
CO-6	2	2	2	3	3	3	2	3	3	3	1	3	**2.5**
CO-7	3	3	3	3	3	3	3	3	3	3	3	3	**3**
Average	**2.7**	**2.7**	**2.6**	**2.4**	**2.7**	**2.9**	**2.3**	**2.7**	**2.7**	**3**	**2.4**	**2.7**	

DETAILED COURSE DESCRIPTIONS

ELECTIVE IV

Cryptography and Network Security

COURSE CODE: 18B1WCI734

COURSE CREDIT: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Introduction to Computers, Knowledge of Computer Networks

Course Objectives:

1. To understand basics of Cryptography and Network Security.
2. To know about various encryption techniques.
3. To be able to secure a message over insecure channel by various means.
4. To learn about how to maintain the Confidentiality, Integrity and Availability of a data.
5. To understand various protocols for network security to protect against the threats in the networks.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Understand the basic security services e.g.Authentication, Access Control, Confidentiality, Integrity, and Non repudiation).	Familiarity
CO-2	Learn standard symmetric encryption algorithms	Assessment
CO-3	Learn the architecture for public and private key cryptography and how public key infrastructure (PKI) supports network security.	Assessment
CO-4	Learn the methods of digital signature and encryption.	Assessment
CO-5	Learn key management and how key exchange protocols work.	Usage
CO-6	Learn futuristic cryptographic techniques like Eliptic Curve and quantum cryptography.	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Block Symmetric Ciphers** Foundation of Security & Cryptography: OSI security architecture, Security Policy, Classical encryption techniques(Substitution Techniques, Transposition Techniques and Staganography) **Mathematical Tools for Cryptography:** Finite fields, number theory,Design Principle of Block Ciphers:DES, Block Cipher Algorithms: AES,Pseudo Random Numbers & Stream Ciphers: Multiple Encryption, Block Cipher modes of operation,stream ciphers, Confidentiality	**2** **6**
2	**Assyymmetric Ciphers** Public Key Cryptography: RSA, Key management,Hashes & Message Digest:Authentication functions,Message authentication codes,Hash functions and their security,Digital Signature, Certificates & standards,Authentication:X.509 Authentication service,	**10**
3	**Security Applications and Protocols** Electronic Mail Security:S/MIME, IP and Web Security	**5**

	Protocols:IPsec, Secure socket layer and transport layer security, secure e-transaction.	
4	**System Security** System Security : Computer Virus, Firewall & Intrusion Detection , Trusted systems, Security Investigation/Audit, Cyber Laws: IT ACT 2000, IT amendment ACT 2008..	**5**
Total lectures		**28**

Suggested Text books:

1. "Cryptography & Network Security" by Stallings, William (Seventh Edition or later).

Other useful resources:

1. Virtual Labs: http://cse29-iiith.virtual-labs.ac.in/index.php?section=Experiments Students are advised to practice virtual lab experiments at above link as and when the topics are covered in the class.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

Course outcomes (Cryptography and Network Security)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	2	1	3	2	2	1	1	1	1	1	1	**1.6**
CO-2	3	2	1	2	2	3	1	3	3	1	2	3	**2.2**
CO-3	2	2	3	1	2	2	3	2	2	3	2	1	**2.1**
CO-4	2	3	3	3	2	3	2	3	2	1	2	1	**2.3**
CO-5	3	2	3	1	2	3	3	2	3	1	2	3	**2.3**
CO-6	3	2	1	1	2	2	2	3	2	1	1	1	**1.8**
Average	**2.7**	**2.2**	**2**	**1.8**	**2**	**2.5**	**2**	**2.3**	**2.2**	**1.3**	**1.7**	**1.7**	

Advanced Algorithms

COURSE CODE: 18B1WCI743

COURSE CREDIT: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Analysis of Data Structures and Algorithms (this pre-requisite will not be waived). You are also expected to have the mathematical maturity to write formal proofs and algorithms.

Course Objectives:

1. Learn to analyze algorithms for Time and Space Complexity
2. Learn asymptotic notations for performance analysis of algorithms.
3. Learn various computing algorithms and data structure used in solving complex problems.
4. Apply important algorithmic design paradigms and method of analysis.
5. Synthesize efficient algorithm design in common engineering design situations.

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	Analyze the asymptotic performance of algorithms.	Familiarity
CO-2	Write rigorous correction proof s of algorithms.	Assessment
CO-3	Demonstrate a familiarity with major algorithms and data structure.	Assessment
CO-4	Apply important algorithmic design paradigms and method of analysis.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Review of Analysis Techniques:** Growth of Functions: Asymptotic notations; Standard notations and common functions; Recurrences and Solution of Recurrence equations- The substitution method, The recurrence – tree method, The master method; Amortized Analysis: Aggregate,Accounting and Potential Methods.	**8**
2	**Graph Algorithms:** Bellman - Ford Algorithm; Single source shortest paths in a DAG; Johnson's Algorithm for sparse graphs; Flow networks and Ford-Fulkerson method; Maximum bipartite matching.	**8**
3	**Polynomials and the FFT:** Representation of polynomials;The DFT and FFT; Efficient implementation of FFT.	**6**
4	**Number -Theoretic Algorithms:** Elementary notions; GCD; Modular Arithmetic; Solving modular linear equations; The Chinese remainder theorem; Powers of an element; RSA cryptosystem; Primality testing; Integer factorization	**6**
5	**String-Matching Algorithms:** Naïve string Matching; Rabin - Karp algorithm; String matching with finite automata; Knuth-Morris-Pratt algorithm; Boyer – Moore algorithms.	**8**

6	**Probabilistic and Randomized Algorithms:** Probabilistic algorithms; Randomizing deterministic algorithms, Monte Carlo and Las Vegas algorithms; Probabilistic numeric algorithms.	**6**
Total lectures		**42**

Suggested Text Book(s):

1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3rd Edition, Prentice-Hall of India, 2010.
2. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.
3. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2nd Edition, Universities press, 2007

Suggested Reference Book(s):

1. S. Muthukrishnan, "Data streams: Algorithms and applications", Foundations and Trends in Theoretical Computer Science, Volume 1, issue 2, 2005.
2. Bach, E., and J. Shallit. Algorithmic Number Theory. Vol. 1. Cambridge, MA: MIT Press, August 26, 1996. ISBN: 9780262024051.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106104019/
2. Link to topics related to course:
   a. https://nptel.ac.in/courses/106104019/1
   b. https://nptel.ac.in/courses/106104019/4
   c. https://nptel.ac.in/courses/106104019/26
   d. https://nptel.ac.in/courses/106104019/27

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Advanced Algorithms)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	2	1.8
CO-2	2	3	3	3	3	1	1	1	2	2	1	2	2
CO-3	2	2	2	2	3	1	1	1	2	2	1	2	1.8
CO-4	2	3	3	3	2	1	1	1	2	3	2	2	2.1
Average	2	2.5	2.5	2.5	2.5	1	1	1	2	2.3	1.5	2	

R Programming

COURSE CODE: 18B1WCI741

COURSE CREDIT: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Basic Programming Language Knowledge, Flow Charts

Course Objectives:

R is open source free software that can handle mathematical and statistical manipulations. R Programming has its own programming language constructs like other languages as well as built in functions to perform any specialized task. This course will cover the concept how to program in R and how to use R for effective data analysis. The students will be able to understand how to install and configure R and how it could be used for an analytics programming environment and gain basic analytic skills via this high-level analytical language. **Upon completion of this course students should be able to:**

1. Introduction and Usages of R Programming
2. How to install R Software and How to use the packages in R Software
3. How to do data management for different applications using R Software
4. Able to draw the Graphs and Plots for better visualization of real life problems.
5. Able to know how the different real applications could be converted according to R Programming Environment for better data analysis.

Course Outcomes:

S. No.	Course outcomes	Level of Attainment
CO-1	To identify the usages of available R packages and associated Open Source software to meet different scientific objectives	Familiarity
CO-2	To understand how to programming in R, reading data into R, accessing R packages,	Assessment
CO-3	Able to write R functions, debugging, profiling R code, and organizing and commenting R code.	Assessment
CO-4	To design and write efficient programs using R to perform routine and specialized data manipulation/management and analysis tasks	Assessment
CO-5	To do data analysis using R for real life applications.	Assessment

Course Contents:

Unit	Contents	Lectures required
1	Introduction and History of R Programming, Basic fundamentals, installation and use of software, data editing, use of R as a calculator, functions and assignments.	4
2	Use of R as a calculator application, functions and matrix operations in R, missing data and logical operators. Conditional executions and loops in R, data management with sequences.	6
3	Data management with repeats, sorting, ordering, and lists Vector indexing, factors, Data management with strings, display and formatting.	6
4	Data management with display paste, split, find and replacement, manipulations with alphabets, evaluation of strings, data frames.	6
5	Data frames, import of external data in various file formats, statistical functions, compilation of data.	4
6	Graphics and plots, statistical functions for central tendency, variation, skewness and kurtosis, handling of bivarite data through graphics, correlations, programming and illustration with examples.	6
7	A Mini Project for Implementation of a Application in R Programming	
Total lectures		**32**

Suggested Text Book(s):

1. Hands-On Programming with R, by Garrett Grolemund, Shroff/O'Reilly; First Edition (2014)
2. Beginning R: The Statistical Programming Language, by Mark Gardener, Wiley (2013)

Suggested Reference Book(s):

1. Benjamin M. Bolker. Ecological Models and Data in R. Princeton University Press, 2008. ISBN 978-0-691-12522-0.
2. Peter Dalgaard. Introductory Statistics with R. Springer, 2nd edition, 2008. ISBN 978-0-387-79053-4.
3. Brian Everitt and Torsten Hothorn. A Handbook of Statistical Analyses Using R. Chapman & Hall/CRC, Boca Raton, FL, 2006. ISBN 1-584-88539-4.
4. John Maindonald and John Braun. Data Analysis and Graphics Using R. Cambridge University Press, Cambridge, 2nd edition, 2007. ISBN 978-0-521-86116-8.
5. Paul Murrell. R Graphics. Chapman & Hall/CRC, Boca Raton, FL, 2005. ISBN 1-584-88486-X.
6. Phil Spector. Data Manipulation with R. Springer, New York, 2008. ISBN 978-0-387-74730-9.
7. W. N. Venables and B. D. Ripley. Modern Applied Statistics with S. Springer, New York, fourth edition edition, 2002.
8. Alain Zuur, Elena N. Ieno, Neil Walker, Anatoly A. Saveiliev, and Graham M. Smith. Mixed Effects Models and Extensions in Ecology with R. Springer, New York, 2009. ISBN 978-0-387-87457-9.
9. Alain F. Zuur, Elena N. Ieno, and Erik Meesters. A Beginner's Guide to R. Use R. Springer, 2009. ISBN: 978-0-387-93836-3.

Other useful resource(s):

1. Link to NPTEL course contents:
   a. https://onlinecourses.nptel.ac.in/noc17_ma17/preview
2. Link to topics related to course:
   a. https://www.coursera.org/learn/r-programming
   b. https://www.edx.org/course/data-science-r-basics
   c. https://www.edx.org/learn/r-programming

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (R Programming)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	1	1	2	1	2	2	2	2	1	1	2	**1.6**
CO-2	3	3	3	3	3	2	2	2	2	1	1	3	**2.3**
CO-3	3	3	3	3	2	2	1	2	3	2	3	3	**2.5**
CO-4	3	3	3	3	2	2	1	2	3	2	3	3	**2.5**
CO-5	3	3	3	3	3	2	2	2	2	1	2	3	**2.4**
Average	**2.8**	**2.6**	**2.6**	**2.8**	**2.2**	**2**	**1.6**	**2**	**2.4**	**1.4**	**2**	**2.8**	

Artificial Intelligence

COURSE CODE: 18B1WCI742

COURSE CREDITS: 2

CORE/ELECTIVE: ELECTIVE

L-T-P: 2-0-0

Pre-requisites: Data Structure, Discrete Structure

Course Objectives:

1. Describe introductory techniques in Artificial Intelligence
2. Heuristic search and adversarial search, Logic for knowledge representation and reasoning
3. Reasoning under uncertainty
4. Machine Learning
5. Apply introductory techniques in Artificial Intelligence to solve realistic problems.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Determine the characteristics of a given problem that an intelligent system must solve.	Familiarity
CO-2	Apply Bayes rule to determine the probability of a hypothesis given evidence.	Assessment
CO-3	Identify examples of knowledge representations for reasoning under uncertainty.	Assessment
CO-4	List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.	Assessment
CO-5	Identify examples of classification tasks, including the available input features and output to be predicted.	Assessment

Course Contents:

Unit	Contents	Lectures Required
1	**Introduction:** Intelligence, Definitions of Intelligent Agents **Single-Agent Search:** Breadth-first, Depth-first and Iterative Deepening Search, Heuristic Search (A* search), Stochastic Local Search (Simulated Annealing, Genetic algorithms)	7
2	**Adversarial Search:** Minimax Search, Alpha-beta pruning, Stochastic Games and Expectiminimax **Knowledge Representation and Logic:** Propositional Logic, Propositional Inference , First-Order Logic, Propositional Inference (Forward chaining, Backward chaining)	7
3	**Reasoning Under Uncertainty:** Probability Bayes Rule , Bayesian Networks, Bayesian Inference	7
4	**Machine Learning**: Definition and examples of broad variety of machine learning tasks, including classification, Inductive learning, Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees, The over-fitting problem, Measuring classifier accuracy	11
Total lectures		**32**

Suggested Text Book(s):

1. Artificial Intelligence a Modern Approach, 3rd Edition. Prentice Hal
2. Artificial Intelligence Hardcover by Elaine Rich and Kevin Knight

Approved in Academic Council held on 28 July 2021

Suggested Reference Book(s):

1. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp by Peter Norvig
2. Machine Learning by Tom M. Mitchell
3. Prediction Machines: The Simple Economics of Artificial Intelligence by Ajay Agrawal , Joshua Gans , Avi Goldfarb

Other useful resource(s):

1. Link to NPTEL course contents:
   - **i.** https://nptel.ac.in/courses/106105077/
   - **ii.** https://nptel.ac.in/courses/106105079/
2. Link to topics related to course:
   - **i.** https://www.ibm.com/developerworks/library/cc-beginner-guide-machine-learning-ai-cognitive/index.html
   - **ii.** https://ai.google/education/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Artificial Intelligence)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	1	1	2	1	2	1	2	2	2	**1.9**
CO-2	3	3	3	1	3	2	1	2	1	2	3	2	**2.2**
CO-3	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO-4	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO-5	3	3	3	3	2	1	2	2	2	2	2	2	**2.3**
Weightage	**3**	**3**	**3**	**2.2**	**2**	**1.4**	**1.6**	**2**	**1.6**	**2**	**2.6**	**2**	

Cryptography and Network Security Lab

COURSE CODE: 18B1WCI774

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisites: Introduction to Computers, Knowledge of Computer Networks

Course Objectives:

1. Be exposed to the different cipher techniques
2. Learn to implement the algorithms like DES, RSA, MD5, SHA-1
3. Understand the Digital Signature Standard
4. Learn to use network security tools like GnuPG, KF sensor, Net Strumbler
5. Be familiar with the intrusion detection system

Course Outcomes:

S.No.	**Course Outcomes**	**Level of Attainment**
CO-1	Implement the cipher techniques	Usage
CO-2	Apply the mathematical foundation required for various cryptographic algorithms	Assessment
CO-3	Develop the various security algorithms	Assessment
CO-4	Design the signature scheme by applying Digital Signature Standard.	Assessment
CO-5	Use different open source tools for network security and analysis	Usage
CO-6	Demonstrate the intrusion detection system	Assessment

List of Experiments:

S.No	**Description**	**Hours**
1	Implementation of Substitution and Transposition Techniques a) Caesar Cipher b) Playfair Cipher c) Hill Cipher d) Vignere Cipher e) Rail Fence Cipher	**3**
2	Implementation of Cryptographic Algorithms a) DES-AES b) RSA Algorithm c) Diffie-Hellman Algorithm	**3**
3	Implementation of Cryptographic Algorithms d) MD5 e) SHA-1	**2**
4	Implement the SIGNATURE SCHEME - Digital Signature Standard(DSS/DSA)	**3**
5	Providing secure data storage, secure data transmission and creating digital signatures	**2**
6	Setup a Honey Pot and Monitor the Honeypot on Network	**2**
7	Installation of toolkits and study the variety of network security options	**2**
8	Perform wireless audit on an access point or a router and decrypt WEP and WPA(Net Stumbler)	**2**

Approved in Academic Council held on 28 July 2021

9	Develop and Demonstrate intrusion detection system	**3**
10	Implement Electronic Mail Security:S/MIME	**3**
11	Implement IPSEC	**4**
Total Lab hours		**28**

Suggested Books/Resources:

1. "Cryptography & Network Security" by Stallings, William (Seventh Edition or later) will be used as the main text book, however the inputs will be supplemented with information from elsewhere wherever the same is required.
2. Virtual Labs: http://cse29-iiith.virtual-labs.ac.in/index.php?section=Experiments Students are advised to practice virtual lab experiments at above link as and when the topics are covered in the class.

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the ProgrammeOutcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO1	3	2	1	3	2	2	1	1	1	1	1	1	**1.6**
CO2	3	2	1	2	2	3	1	3	3	1	2	3	**2.2**
CO3	2	2	3	1	2	2	3	2	2	3	2	1	**2.1**
CO4	2	3	3	3	2	3	2	3	2	1	2	1	**2.3**
CO5	3	2	3	1	2	3	3	2	3	1	2	3	**2.3**
C06	2	2	2	1	2	2	2	3	2	1	1	1	**1.8**
Average	**2.5**	**2.2**	**2.2**	**1.8**	**2**	**2.5**	**2**	**2.3**	**2.2**	**1.3**	**1.7**	**1.7**	

Advanced Algorithms Lab

COURSE CODE: 18B1WCI773

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisites: None

Course Objectives:

1. Learn to analyze algorithms for Time and Space Complexity
2. Learn asymptotic notations for performance analysis of algorithms.
3. Learn various computing algorithms and data structure used in solving complex problems.
4. Apply important algorithmic design paradigms and method of analysis.
5. Synthesize efficient algorithm design in common engineering design situations.
6. Design Bellman-Ford algorithm and determine its performance.
7. Design a Miller Rabin algorithm and Monte Carlo algorithm to test the primality of a given integer and determine its performance.
8. Design a string matching problems.

Course outcomes:

S.NO	**Course outcomes**	**Level of Attainment**
CO-1	Analyze the asymptotic performance of algorithms.	Familiarity
CO-2	Write rigorous correction proof s of algorithms.	Assessment
CO-3	Demonstrate a familiarity with major algorithms and data structure.	Assessment
CO-4	Apply important algorithmic design paradigms and method of analysis.	Usage

List of Experiments:

S.No	**Description**	**Hours**
1	Design, develop a program to implement the Bellman-Ford algorithm and determine its performance.	**4**
2	Design, develop a program to implement a Miller Rabin algorithm to test the primality of a given integer and determine its performance.	**4**
3	Design, develop a program to implement a Monte Carlo algorithm to test the primality of a given integer and determine its performance.	**4**
4	Design, develop a program to solve the string matching problem using naïve approach and the KMP algorithm and compare their performances.	**4**
5	Design, develop a program to solve string matching problem using Finite Automata and determine its performance.	**6**
6	Design, develop and write program to solve string matching problem using Robin Karp algorithm and determine its performance.	**6**
Total Lab hours		**28**

Suggested Books/Resources:

1. T. H Cormen, C E Leiserson, R L Rivest and C Stein: Introduction to Algorithms, 3rd
2. Edition, Prentice-Hall of India, 2010.
3. Kenneth A. Berman, Jerome L. Paul: Algorithms, Cengage Learning, 2002.
4. Ellis Horowitz, Sartaj Sahni, S.Rajasekharan: Fundamentals of Computer Algorithms, 2nd

5. Edition, Universities press, 2007
6. S. Muthukrishnan, "Data streams: Algorithms and applications", Foundations and Trends in Theoretical Computer Science, Volume 1, issue 2, 2005.
7. Bach, E., and J. Shallit. Algorithmic Number Theory. Vol. 1. Cambridge, MA: MIT Press, August 26, 1996. ISBN: 9780262024051.
8. Link to topics related to course:
   i. https://tejaswinihbhat.blogspot.com/2016/07/program-1-bellman-ford-algorithm-design.html
   ii. https://www.sanfoundry.com/c-program-implement-rabin-miller-primality-test-check-number-prime/
   iii. https://www.cs.bu.edu/fac/lnd/toc/z/node21.html
   iv. http://cs.indstate.edu/~kmandumula/abstract.pdf

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	2	**1.8**
CO-2	2	3	3	3	3	1	1	1	2	2	1	2	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	2	**1.8**
CO-4	2	3	3	3	2	1	1	1	2	3	2	2	**2.1**
Average	**2**	**2.5**	**2.5**	**2.5**	**2.5**	**1**	**1**	**1**	**2**	**2.3**	**1.5**	**2**	

Approved in Academic Council held on 28 July 2021

R Programming Lab

COURSE CODE: 18B1WCI771

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisites: None

Course Objectives:

1. Understand the Usages of R Programming, Installation of R, Configure and Execute Program.
2. Use of different packages in R Software for different problems
3. Usages of Functions and Classes in R Software
4. Able to draw the Graphs and Plots for better visualization of real life problems using plot methods.
5. Able to understand how the R Programming could be used for different case studies.

Course Outcome (CO)

S.No.	Course outcomes	Level of Attainment
CO-1	To identify the usages of methods and classes in R to meet different scientific objectives	Familiarity
CO-2	To identify the usages of plot functions in R to represent data in better form	Familiarity
CO-3	Usage of R packages, How to download new packages and use them in R.	Computational skills
CO-4	To do data analysis tasks on sample data set using R Programming.	Technical skills
CO-5	To do data analysis and exploration using R for real life applications. How it could be used further in research and analysis on datasets.	Computational skills & Assessment

List of Experiments:

S.No	Description	Hours
1	**What is R? How to Install R :- Download, Install and Configure**	2
2	**Basics of R Programming:-** Algebra, Vectors, Matrices, Manipulation, Loops/Statements	2
3	**Data Types in R Programming:-** Data Types, Converting/Using Data Types in R Programming	2
4	**Reading in Data:-** Types of Input, How to Read In Data,	2
5	**Plotting Data:-** Dot Plots, Histograms, Box Plots, Additional Features in Plotting Data in R	2
6	**Exporting Data in R:-** Types of Output, How to Export Data	2
7	**Exporting Data in R:-** Types of Output, How to Export Data	2
8	**Functions in R Programming:-** Built In Functions in R, Custom Functions in R, Graphical Functions in R	2

9	**Functions in R Programming:-** Built In Functions in R, Custom Functions in R, Graphical Functions in R	**2**
10	**Tips for Writing Good R Code:-** General Practices, Matrix Multiplication, Packages in R, Usage of Packages in R, Help in R	**2**
11	**R Editors:-** Built In R Editors, Other Editors in R, **Measures of Central Tendency and Dispersion**	**2**
12	**Statistical analyses with R:-** A simple example of analysis of variance, Array and Strings in R, Hypothesis Testing: Testing the Significance of the Difference Between Two Means	**2**
13	**Classes and Methods in R Programming:-** Setting Classes and Methods, Different Usages of Classes and Methods, Bivariate Statistics for Nominal Data	**2**
14	Code for Sample Case Study Using Methods in R Programming, List of Most Useful Functions in R Programming, Bivariate Statistics for Ordinal Data, Bivariate Statistics for Interval/Ratio Data	**2**
Additional Exercise-	Case Study for Data Analysis Using R Programming	**---**
Total Lab hours		**28**

Suggested Books/Resources:

1. Hands-On Programming with R, by Garrett Grolemund, Shroff/O'Reilly; First Edition (2014)
2. Beginning R: The Statistical Programming Language, by Mark Gardener, Wiley (2013)
3. Peter Dalgaard. Introductory Statistics with R. Springer, 2nd edition, 2008. ISBN 978-0-387-79053-4.
4. John Maindonald and John Braun. Data Analysis and Graphics Using R. Cambridge University Press, Cambridge, 2nd edition, 2007. ISBN 978-0-521-86116-8.
5. Alain F. Zuur, Elena N. Ieno, and Erik Meesters. A Beginner’s Guide to R. Use R. Springer, 2009. ISBN: 978-0-387-93836-3.
6. Link to NPTEL course contents:
   https://onlinecourses.nptel.ac.in/noc17_ma17/preview
7. Important URLs:
   R Manuals: http://cran.r-project.org/ > Documentation > Manuals
   R Journal: http://journal.r-project.org/
   R Forum: http://www.nabble.com/R-f13819.html
   CRAN Home Page: http://cran.r-project.org/
8. Link to topics related to course:
   a. https://www.coursera.org/learn/r-programming
   b. https://www.edx.org/course/data-science-r-basics
   c. https://www.edx.org/learn/r-programming
   d. http://web.math.ku.dk/~helle/R-intro/exercises.pdf
   e. https://afit-r.github.io/basics
   f. https://www.cs.upc.edu/~robert/teaching/estadistica/rprogramming.pdf
   g. https://faculty.washington.edu/tlumley/Rcourse/R-fundamentals.pdf

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	2	1	2	2	3	2	1	3	2	1	**2**
CO-2	3	3	2	1	1	3	3	1	3	3	3	1	**2.3**
CO-3	3	3	2	1	2	2	2	1	2	3	3	2	**2.2**
CO-4	3	3	3	2	2	2	3	2	2	1	2	2	**2.3**
CO-5	3	3	2	2	2	2	2	2	2	3	3	3	**2.4**
Average	**3**	**2.8**	**2.2**	**1.4**	**1.8**	**2.2**	**2.6**	**1.6**	**2**	**2.6**	**2.6**	**1.8**	

Artificial Intelligence Lab

COURSE CODE: 18B1WCI772

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

Pre-requisite: None

Course Objectives:

1. Describe introductory techniques in Artificial Intelligence
2. Heuristic search and adversarial search, Logic for knowledge representation and reasoning
3. Reasoning under uncertainty
4. Machine Learning
5. Apply introductory techniques in Artificial Intelligence to solve realistic problems.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Determine the characteristics of a given problem that an intelligent system must solve.	Familiarity
CO-2	Apply Bayes' rule to determine the probability of a hypothesis given evidence.	Assessment
CO-3	Identify examples of knowledge representations for reasoning under uncertainty.	Assessment
CO-4	List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised.	Assessment
CO-5	Identify examples of classification tasks, including the available input features and output to be predicted.	Assessment

List of Experiments:

S.No	Description	Hours
1	Implementation of DFS and BFS Searching Algorithms	**2**
2	Implementation of A* Algorithm	**2**
3	Study of Prolog Language	**2**
4	Write simple fact for the statements using PROLOG.	**2**
5	Write predicates One converts centigrade temperatures to Fahrenheit, the other checks if a temperature is below freezing	**2**
6	WAP in turbo prolog for medical diagnosis and show that advantage and disadvantage of green and red cuts.	**2**
7	Write a program to solve the Monkey Banana problem	**2**
8	Study of LISP Language	**2**
9	Write a program to solve water jug problem using LISP	**2**
10	Consider House-votes dataset provided in lab. The task is to predict whether the voter is a republican or a democrat based on their votes using Naive Bayes algorithm with 5-fold cross validation. It has 16 binary attributes and 2 classes.	**2**
11	In Experiment 10, Estimate the accuracy of Naive Bayes algorithm using 5-fold cross validation on the house-votes data set.	**2**

Approved in Academic Council held on 28 July 2021

<table>
<tr><td>12</td><td>In Experiment 10, Estimate the precision, recall, accuracy, and F-measure using 10-fold cross-validation.</td><td><b>2</b></td></tr>
<tr><td>13</td><td>Consider Breast Cancer data set provided in class. It has 9 numeric attributes and 2 types of cancer to be predicted. Compare the performance of 10 machine learning models for given classification data set for the data partition of 70-30%.
<table>
<tr><th>Model</th><th>Sensitivity</th><th>Specificity</th><th>Precision</th><th>Recall</th><th>Accuracy</th><th>F-Score</th></tr>
<tr><td>M1</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>M2</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>M10</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td><b>2</b></td></tr>
<tr><td>14</td><td>Ensemble the models from Experiment 13 for given data set on data partition of 70-30%.
<table>
<tr><th>Model</th><th>Combination</th><th>Sensitivity</th><th>Specificity</th><th>Precision</th><th>Recall</th><th>Accuracy</th></tr>
<tr><td>E1</td><td>M1, M5, M6, M7, M10</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>E2</td><td>M1, M2, M4</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>E3</td><td>M2, M4, M6, M8, M10</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>E4</td><td>M5, M7, M8</td><td></td><td></td><td></td><td></td><td></td></tr>
</table></td><td><b>2</b></td></tr>
<tr><td colspan="2"><b>Total Lab hours</b></td><td><b>28</b></td></tr>
</table>

Suggested Books/Resources:

1. Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp by Peter Norvig
2. Programming in Prolog-Springer by William F. Clocksin, Christopher S. Mellish
3. Machine Learning by Tom M. Mitchell
4. Prediction Machines: The Simple Economics of Artificial Intelligence by Ajay Agrawal , Joshua Gans , Avi Goldfarb
5. Artificial Intelligence a Modern Approach, 3rd Edition. Prentice Hal
6. Artificial Intelligence Hardcover by Elaine Rich and Kevin Knight
7. Link to topics related to course:
   - **i.** https://www.ibm.com/developerworks/library/cc-beginner-guide-machine-learning-ai-cognitive/index.html
   - **ii.** https://ai.google/education/

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	3	3	2	1	2	1	2	1	2	2	2	**2**
CO-2	3	3	3	2	3	2	1	2	1	2	3	2	**2.3**
CO-3	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO-4	3	3	3	3	2	1	2	2	2	2	3	2	**2.3**
CO-5	3	3	3	3	2	1	2	2	2	2	2	2	**2.3**
Average	**3**	**3**	**3**	**2.6**	**2**	**1.4**	**1.6**	**2**	**1.6**	**2**	**2.6**	**2**	

DETAILED COURSE DESCRIPTIONS

ELECTIVE V

Approved in Academic Council held on 28 July 2021

Storage Networks

COURSE CODE: 18B1WCI736

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: None

Course Objectives:

1. To learn the ability to design and implement the various aspects of storage networks.
2. To know the design model SAN, NAS, DAS, CAS, etc.
3. To learn the various technologies like SCSI, Fibre Channel, INFINIBAND, IP Stoarge etc.
4. To learn the concepts of virtualization.
5. To learn the protocols of Distributed storage networks.

Course outcomes:

S.NO	Course Outcomes	Level of Attainment
CO-1	Basics of Storage networks	Familiarity
CO-2	Design and Implement the RAID Levels.	Assessment
CO-3	Discussing algorithms related to storage networks.	Assessment
CO-4	Designing the SAN, NAS, CAS, and DAS based storage networks	Assessment
CO-5	Concepts of Data Deduplication, and File Systems.	Assessment
CO-6	Discussion of advanced topics of Distributed storage networks, protocols, and architecture.	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Intorduction to Storage Technology:** Data proliferartion, Overview of storage infrastructure components, Evolution of storage, Information Lifecycle Management concept, Basic storage management skills and activities.	6
2	**Technologies for Storage Networks:** Disk Subsystems, Overview Architecture of Intelligent Disk Subsytem, JBOD: Just A Bunch Of Disks, RAID & RAID Levels, Hot Sparing, Hard Disks and Internal I/O Channels, Caching: Acceleration of Hard Disk Access.	8
3	**I/O Techniques:** DAS, SAN, NAS, evolution, Storage Area Networks (SAN): elements & connectivity, Fibre Channel SAN & Products, IP SAN Technology & Products, IP SAN elements, standards (iSCSI, iFCP, mFCP, FCIP and iSNS), Migration from SCSI and Fibre Channel to IP storage, Network attached Storage: elements & connectivity.	10
4	**Management of Storage Network:** Requirements of Management Systems, Management Interfaces Standardized and Proprietary Mechanisms, In-	6

	band & Out- band Management.	
5	**Storage Virtualization:** The concept of storage virtualization, Storage virtualization on various levels of the storage network, Symmetric & Asymmetric Storage virtualization, Performance of SAN virtualization, Scaling storage with virtualization.	**7**
6	**Distributed Storage Networks:** Architecture, Protocols Applications, Data Deduplications, File Systems	**5**
Total lectures		**42**

Suggested Text Book(s):

1. **Storage Networks Explained"**, Ulf Troppens, Rainer Erkens, ISBN 0-470-86182-7, John Wiley& Sons.

Suggested Reference Book(s):

1. "Storage Networks: The Complete Reference", R. Spalding, ISBN:0072224762, McGraw-Hill
2. "Storage Networking Fundamentals: An Introduction to Storage Devices, Subsystems, Applications, Management, and Filing Systems", Marc Farley, ISBN: 1-58705-162-1, Cisco Press.

3. "Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs, Second Edition", Tom Clark, ISBN: 0-321-13650-0, Addison Wesley

Evaluation Scheme:

S.No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 hour.	Syllabus covered upto Test-1
2	T-2	25	1.5 hours	Syllabus covered upto Test-2
3	T-3	45	2 hours	Syllabus covered upto Test-3
3.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Storage networks)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	1	3	1	3	**2.4**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.3**	**3**	**1**	**3**	

Internet of Things

COURSE CODE: 18B1WCI738

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: Programming experience, preferably in MATLAB, and/or C/C++/C#/Python/Java

Course Objectives:

1. Understand the basics of IoT, The various IoT protocols and applications of various IoT technologies.
2. Be familiar with the basics of Contiki OS and standardized protocols for IPv6 Low power networking
3. Know the IEEE 802.15.4 standard and 6 LoWPAN technology
4. Understand the RPL protocol and to understand the various routing Issues in IoT
5. Be exposed to the security issues, data collection and distributed computing

Course outcomes:

S.NO	Course outcomes	Level of Attainment
CO-1	To learn IoT Architecture, Security Concerns, Security challenges and Issues, IoT five layers Security at different layers, The IoT protocols	Familiarity
CO-2	To learn about Contiki Operating System, The Hardware Platforms, IP networking, The standardized protocols for IPv6 Low power networking, The COOJA network simulator.	Familarity
CO-3	To learn The IPv6 packet format, IEEE 802.15.4 standard, IPv6 over IEEE 802.15.4, 6LoWPAN packet format, Addressing, Forwarding and Routing, Header compression, Fragmentation and Reassembly, Multicasting.	Assessment
CO-4	To study the Routing Issues in IoT, The RPL protocol and Distance Vector Routing, Storing and Non-Storing Mode of Operation, RPL control messages (ICMPv6) i.e, DIO-DAO-DIS, The objective function, Loop detection and Repair mechanisms, RPL implementation with COOJA simulator	Assessment
CO-5	To study the various development boards and their interfacing with IoT	Assessment
CO-6	To learn about Security Issues in RPL, Data collection for IoT applications, Distributed Computing in IoT	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Internet-of-Things (IoT)** Introduction to IoT, The IoT Architecture, Security Concerns, Security challenges and Issues, Security at different layers, The IoT protocols	4
2	**Contiki Operating System** The Overview of Contiki Operating System, The Hardware Platforms, IP networking, The standardized protocols for IPv6 Low power networking, The COOJA network simulator	4
3	**IEEE Standards** The IPv6 packet format, IEEE 802.15.4 standard, IPv6 over IEEE 802.15.4, 6LoWPAN	12

	packet format, Addressing, Forwarding and Routing, Header compression, Fragmentation and Reassembly, Multicasting	
4	**Routing** Routing Issues in IoT, The RPL protocol and Distance Vector Routing, Storing and Non-Storing Mode of Operation, RPL control messages (ICMPv6) i.e, DIO-DAO-DIS, The objective function, Loop detection and Repair mechanisms, RPL implementation with COOJA simulator	12
5	**Security Issues** Security Issues in RPL, Data collection for IoT applications, Distributed Computing in IoT	10
Total Lectures		**42**

Suggested Text Book(s):

i. Internet-of-Things (IoT) Systems Architectures, Algorithms, Methodologies, Dimitrios Serpanos and Marilyn Wolf, 1st edition, Springer, 2017.

ii. Building The Internet-of-Things, Maciej Kranz, Wiley, 2016.

Suggested Reference Book(s):

iii. Learning Internet-of-Things, Peter Waher, Packt Publisher, 2015.

iv. IoT in 5 days: an easy guide to Wireless Sensor Networks (WSN), IPv6 and the Internet-of-Things(IoT) , Antonio Linan Colina , Alvaro Vives, Antoine Bagula, Marco Zennaro and Ermanno Pietrose, 2015.

Other useful resource(s):

Link to NPTEL course contents: https://nptel.ac.in/courses/106105166/

Link to topics related to course:

i. https://nptel.ac.in/courses/106105166/1
ii. https://nptel.ac.in/courses/106105166/6
iii. https://nptel.ac.in/courses/106105166//3
iv. https://nptel.ac.in/courses/106105166//10
v. https://nptel.ac.in/courses/106105166//12
vi. https://nptel.ac.in/courses/106105166//15
vii. https://nptel.ac.in/courses/106105166/21
viii. https://nptel.ac.in/courses/106105166//29

Evaluation Scheme:

S.No	Exam	Marks	Duration	Coverage/Scope of Examination
1	T-1	15	1 hr.	Syllabus covered up to Test- 1.
2	T-2	25	1.5 hr.	Syllabus covered up to Test- 2.
3	T-3	35	2 hr.	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course Outcomes (Internet of Things)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	1	3	1	3	**2.4**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.3**	**3**	**1**	**3**	

Mobile Computing

COURSE CODE: 18B1WCI735

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites (if any): NIL

Course Objectives:

1 To learn about the concepts and principles of mobile computing;
2 To explore both theoretical and practical issues of mobile computing;
3 To develop skills of finding solutions and building software for mobile computing applications.

Course Outcome (CO)

S.NO	Course outcomes	Level of Attainment
CO-1	Overview of mobile computing: Motivations, concepts, challenges and applications of mobile computing, relationship with distributed computing, internet computing, ubiquitous/pervasive computing, mobile computing models and architectures	Familiarity
CO-2	Wireless networks: Wireless communication concepts, classification of wireless networks, cellar networks (1G, 2G, 3G, 4G), WLAN, WPAN, WMAN, satellite networks	Usage
CO-3	Mobile device plateforms: Mobile devices, mobile OS, J2ME, Windows mobile and .Net framework, BREW	Assessment
CO-4	Wireless Mobile Internet Wireless Internet architecture; Wireless gateway; Wireless application server; Synchronization server; Messaging server; Mobile Internet proxy services (transcoding, caching); Data dissemination; Disconnected operations (hording)	Assessment
CO-5	Mobile ad hoc networks: Concepts and applications; routing in mobile ad hoc networks; sensor networks, mobile peer-to-peer computing	Assessment
CO-6	Mobility management: Handoff and location management concepts; mobility management in PLMN; mobility management in mobile Internet; mobility management in mobile agent systems; adaptive location management methods	Assessment
CO-7	Location-based services: LBS applications; mobile positioning techniques; GIS; LBS architecture and protocols	Assessment
CO-8	Mobile device technology: Mobile app programming, QR Code applications, Simple software development tools for mobile apps	Usage

Course Contents:

Unit	Topic	Lectures required
1	Overview of mobile computing: Motivations, concepts, challenges and applications of mobile computing, relationship with distributed computing, internet computing, ubiquitous/pervasive computing, mobile computing	**03**

	models and architectures	
2	Wireless networks: Wireless communication concepts, classification of wireless networks, cellar networks (1G, 2G, 3G, 4G), WLAN, WPAN, WMAN, satellite networks	**09**
3	Mobile device plateforms: Mobile devices, mobile OS, J2ME, Windows mobile and .Net framework, BREW	**09**
4	Wireless Mobile Internet Wireless Internet architecture; Wireless gateway; Wireless application server; Synchronization server; Messaging server; Mobile Internet proxy services (transcoding, caching); Data dissemination; Disconnected operations (hording)	**01**
5	Mobile ad hoc networks: Concepts and applications; routing in mobile ad hoc networks; sensor networks, mobile peer-to-peer computing	**06**
6	Mobility management: Handoff and location management concepts; mobility management in PLMN; mobility management in mobile Internet; mobility management in mobile agent systems; adaptive location management methods	**04**
7	Location-based services: LBS applications; mobile positioning techniques; GIS; LBS architecture and protocols	**03**
8	Mobile device technology: Mobile app programming, QR Code applications, Simple software development tools for mobile apps	**02**
	Total lectures	**42**

Suggested Textbooks

1. R. Meier, Professional Android application development. Indianapolis, IN: Wiley, 2009.
2. Microsoft Open Technologies, Inc., Windows Phone 8 Guide for Android Application Developers.
3. Stalling, William, 2002. Wireless Communications and Networks. 2nd ed. Upper Saddle River, NJ 07458: Pearson

Suggested Reference books:

1 W. Lee and K. Mittal, Beginning Android application development. Indianapolis, Ind.: Wiley Pub., 2011.

2 A. Whitechapel and S. McKenna, Windows Phone 8 development internals. Redmond, Wash.: Microsoft Press, 2012.

3 I. Stojmenovic, Handbook of Wireless Networks and Mobile Computing. Hoboken, NJ: John Wiley & Sons, 2002.

4 Dr.S.S.Dhenakaran, A.Parvathavarthini (2013) 'An Overview of Routing Protocols in Mobile Ad-Hoc Network', International Journal of Advanced Research in Computer Science and Software Engineering, 3(2), pp. [Online]. Available at: www.ijarcsse.com.

Evaluation Scheme

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course Outcomes (Mobile Computing)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	2	3	2	2	3	3	3	3	3	1	3	**2.6**
CO-2	3	3	3	2	2	3	3	3	3	3	1	3	**2.7**
CO-3	3	3	3	3	3	3	3	3	3	3	1	3	**2.8**
CO-4	3	3	3	2	3	3	3	3	3	3	1	3	**2.8**
CO-5	3	3	3	3	2	3	3	3	2	3	1	3	**2.7**
CO-6	2	3	3	3	3	2	3	3	3	3	1	3	**2.7**
CO-7	3	2	2	3	3	3	3	3	2	2	2	2	**2.5**
CO-8	3	2	2	2	2	3	3	3	3	2	2	2	**2.4**
Average	**2.9**	**2.6**	**2.8**	**2.5**	**2.5**	**2.9**	**3**	**3**	**2.8**	**2.8**	**1.3**	**2.8**	

Cloud Computing

COURSE CODE: 18B1WCI737

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: Operating System

Course Objectives:

1. To demonstrate an understanding of cloud computing concepts and standards.
2. To understand all enabling technologies of Cloud computing.
3. To discuss issues and challenges pertaining to management of emerging cloud computing technologies and learn approaches to manage them.
4. To demonstrate the practical implementation and usage scenarios of Cloud computing.

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	To learn the basic concepts, applications and terminology of cloud computing.	Familiarity
CO-2	To understand different enabling technologies for Cloud computing environment.	Assessment
CO-3	To design Cloud computing data-center for effective utilization of available resources	Usage
CO-4	To study different managers related to Cloud computing services	Assessment
CO-5	To understand dofferent case studies of Cloud computing and its advance topics	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Understanding Cloud Computing:** Basic Concepts and terminology, Goals and Benefits. Risks and Challenges, Roles and boundaries, Cloud characteristics, Cluster Computing, Grid Computing, NIST Architecture, Cloud Deployment models, Cloud service models.	7
2	**Cloud Enabling Technologies:** Virtualization, Types of virtualization, Server Consolidation, virtualization management, Web Technology, Service, Oriented Architecture, Datacenter and Multi-tenancy	7
3	**Cloud Infrastructure Management:** Cloud datacenter design, Workloads and software infrastructure for a datacenter, Datacenter hardware, energy and power efficiency in a datacenter., Cloud usage monitor, Monitoring agent, Resource agent, Polling Agent	7
4	**Cloud Mechanisms:** Automated Scaling, Load Balancer, SLA Monitor, Failover System, Multi-Cloud Broker	7

5	**Fundamental Cloud Architectures:** workload distribution architecture, resource pooling architecture, dynamic scalability architecture, service load balancing architecture, cloud brusting architecture. Billing Management System Business cost metrics, cloud usage cost metrics,	**7**
6	Cloud service metrics., Cloud Security, Mobile cloud computing, Disaster recovery in cloud computing, Case studies	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Cloud Computing: Concepts, Technology & Architecture, by Zaigham Mahmood, Thomas Erl, Ricardo Puttini, Prentice Hall, ISBN: 9780133387568
2. Cloud Computing Bible, by Barrie Sosinsky, Barrie Sosinsky.

Suggested Reference Book(s):

1. Cloud Computing: A Practical Approach by Anthony T. Velte, Toby J. Velte and Robert Elsenpeter; Tata McGraw – Hill Edition
2. The Datacenter as a Computer – An Introduction to the Design of Warehouse Scale Machines by Luiz Andre Barroso and Urs Holzle; Morgan and Claypool Publishers

3. Cloud Computing Explained: Implementation Handbook for Enterprises by John Rhoton
4. The Cloud at Your Service by Jothy Rosenburg and Arthur Mateos

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc17_cs23/preview
2. Link to topics related to course:
   i. https://www.edx.org/learn/cloud-computing
   ii. https://www.udemy.com/introduction-to-cloud-computing/

Evaluation Scheme:

S.No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Tutorials / Assignments, Quizzes, Attendance	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Cloud Computing)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	1	1	1	2	2	1	2	2	1	2	3	**1.8**
CO-2	3	2	1	2	2	2	1	2	2	1	3	2	**1.9**
CO-3	2	2	3	2	2	1	3	2	1	1	1	1	**1.8**
CO-4	2	2	3	2	2	1	3	2	1	1	1	1	**1.8**
CO-5	3	3	3	3	3	1	2	1	2	2	3	2	**2.3**
Average	**2.6**	**2**	**2.2**	**2**	**2.2**	**1.4**	**2**	**1.8**	**1.6**	**1.2**	**2**	**1.8**	

Approved in Academic Council held on 28 July 2021

Computational Techniques and Algorithms in Engineering

COURSE CODE: 18B1WCI740

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: A basic background in engineering mathematics and computational techniques is assumed. In particular, it is assumed that the student has a basic understanding of linear algebra, probability theory, on which the more advanced material in this course will be built.

Course Objectives:

1. To Solve systems of linear equations using multiple methods, including Gaussian elimination and matrix inversion. Algorithms for vector and matrix operations.
2. Positive Definite Systems, Cholesky Decomposition, LU Decomposition, Sensitivity and round-off errors.
3. To discuss Least Squares Problem, OR Decomposition.
4. To determine Eigen values and eigenvectors and solve Eigen value problems.

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	Solve systems of linear equations using multiple methods, including Gaussian elimination and matrix inversion. Algorithms for vector and matrix operations.	Usage
CO-2	Positive Definite Systems, Cholesky Decomposition, LU Decomposition, Sensitivity and round-off errors	Usage
CO-3	Least Squares Problem, OR Decomposition	Usage
CO-4	SVD and QR algorithm	Usage
CO-5	,Determine Eigen values and eigenvectors and solve Eigen value problems	Assessment
CO-6	Iterative algorithms and Convergence	Assessment

Course Contents:

Unit	Contents	Lectures required
1	Matrix Vector and Matrix Matrix Multiplication Algorithms, Gaussian Elimination and Its Variants, Systems of Linear Equations, Triangular Systems, Positive Definite Systems, Cholesky Decomposition, Banded Positive Definite Systems, Sparse Positive Definite Systems, Gaussian Elimination and the LU Decomposition, Gaussian Elimination with Pivoting, Sparse Gaussian Elimination	5
2	Sensitivity of Linear Systems,Vector and Matrix Norms, Condition Numbers, Perturbing the Coefficient Matrix, A Posteriori Error Analysis Using the Residual, Roundoff Errors, Backward Stability, Propagation of Roundoff Errors, Backward Error Analysis of Gaussian Elimination,Scaling, Componentwise Sensitivity Analysis.	6
3	The Least Squares Problem, The Discrete Least Squares Problem, Orthogonal Matrices, Rotators, and Reflectors, Solution of the Least Squares Problem, The	8

	Gram-Schmidt Process, Geometric Approach, Updating the QR Decomposition.	
4	The Singular Value Decomposition, Some Basic Applications of Singular Values,The SVD and the Least Squares Problem, Sensitivity of the Least Squares Problem.	**4**
5	Eigenvalues and Eigenvectors, Systems of Differential Equations, The Power Method and Some Simple Extensions, Similarity Transforms, Reduction to Hessenberg and Tridiagonal Forms, The QR Algorithm, Implementation of the QR algorithm, Use of the QR Algorithm to Calculate Eigenvectors.	**12**
6	Iterative Methods for Linear Systems, The Classical Iterative Methods, Convergence of Iterative Methods, Descent Methods; Steepest Descent, Preconditioners, The Conjugate- Gradient Method, Derivation of the CG Algorithm, Convergence of the CG Algorithm, Indefinite and Nonsymmetric Problems.	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Fundamentals of Matrix Computations, DAVID S. WATKINS
2. Linear Algebra Done Right, by Sheldon Axler

Suggested Reference Book(s):

1. Gilbert Strang, Linear Algebra and Its Applications, 4th Edition, Brooks Cole, 2006.
2. Gene H. Golub and Charles F. Van Loan, Matrix Computations, 3rd edition, John Hopkins University Press, 1996, ISBN 0-8018-5414-8.
3. Lloyd N. Trefethen and D. Bau III, Numerical Linear Algebra, SIAM, 1997.
4. James W. Demmel, Applied Numerical Linear Algebra, SIAM, 1997
5. Anne Greenbaum, Iterative Methods for Solving Linear Systems, SIAM, 1997.
6. Yousef Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.
7. William L. Briggs, Van Emden Henson, Steve F. McCormick, A Multigrid Tutorial, 2nd edition, SIAM, 2000.
8. B.W. Kernighan, D.M. Ritchie, C Programming Language (2nd edition). Prentice Hall, 1988.
9. M. Banahan, D. Brady and M. Doran, The C Book, second edition, Addison Wesley, 1991.
10. C Programming, Wikibooks

Other useful resource(s):

Evaluation Scheme:

S.No	**Exam**	**Marks**	**Duration**	**Coverage / Scope of Examination**
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Tutorials / Assignments, Quizzes, Attendance	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Advanced Computational Techniques in Engineering)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**

CO-1	3	2	3	2	2	3	2	3	2	2	3	3	83.3
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	86.1
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	80.6
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	83.3
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	80.6
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	86.1
Average	88.9	72.2	88.9	72.2	83.3	83.3	83.3	83.3	72.2	88.9	94.4	88.9	

DETAILED COURSE DESCRIPTIONS

ELECTIVE VI

Data Analytics

COURSE CODE: 18B1WCI843

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: Linear algebra, calculus, probability theory and statistics

Course Objectives:

Data Analytics is the science of analyzing data to convert information to useful knowledge. This knowledge could help us understand our world better, and in many contexts enable us to make better decisions. While this is the broad and grand objective, the last 20 years has seen steeply decreasing costs to gather, store, and process data, creating an even stronger motivation for the use of empirical approaches to problem solving.

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	Gaining factual knowledge regarding data acquisition, data cleansing, and various aspects of data analytics and visualization	Familiarity
CO-2	Learning the principles of data analytics and its underlying methods and algorithms	Assessment
CO-3	Learning to apply the methods of data collection and data analytics to solve business and related problems in support of business decision- making	Assessment
CO-4	Developing the skills necessary to use related software tools to perform data collection, cleansing, and analytics	Usage

Course Contents:

Unit	Contents	Lectures required
1	Introduction to the course, Descriptive Statistics, Probability Distributions	5
2	Inferential Statistics through hypothesis tests, Permutation & Randomization Test	4
3	Regression, ANOVA(Analysis of Variance)	5
4	Differentiating algorithmic and model based frameworks Regression : Ordinary Least Squares, Ridge Regression, Lasso Regression, K Nearest Neighbours Regression & Classification	7
5	Bias-Variance Dichotomy, Model Validation Approaches Logistic Regression, Linear Discriminant Analysis Quadratic Discriminant Analysis Regression and Classification Trees Support Vector Machines	8
6	Ensemble Methods: Random Forest, Neural Networks, Deep learning	4
7	Clustering, Associative Rule Mining, Challenges for big data anlalytics	4
8	Creating data for analytics through designed experiments, Creating data for analytics through Active learning Creating, data for analytics through Reinforcement learning	5
Total lectures		**42**

Approved in Academic Council held on 28 July 2021

Suggested Reference Book(s):

1 Hastie, Trevor, et al. The elements of statistical learning. Vol. 2. No. 1. New York: Springer, 2009.

2. Montgomery, Douglas C., and George C. Runger. Applied statistics and probability for engineers. John Wiley & Sons, 2010.

Other useful resource(s):

1. Link to NPTEL course contents: https://onlinecourses.nptel.ac.in/noc15_mg05/preview

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Data Analytics)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	2	2	2	2	1	1	1	2	2	2	2	**1.8**
CO-2	2	3	3	3	3	1	1	1	2	2	1	2	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	2	**1.8**
CO-4	2	3	3	3	2	1	1	1	2	3	2	2	**2.1**
Average	**2**	**2.5**	**2.5**	**2.5**	**2.5**	**1**	**1**	**1**	**2**	**2.3**	**1.5**	**2**	

Big Data

COURSE CODE: 18B1WCI844

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: **None**

Course Objectives:

1. To learn the basic concepts and terminology in big data analytics
2. To learn about the map reduce and the new software stack
3. To learn about the mining of data streams, estimating moments and windowing, link analysis: page rank and efficient computation of page rank
4. To learn concepts associated with frequent item sets from big data and counting frequent items from stream
5. To learn about clustering for big data and mining of social network graph
6. To learn about recommendation systems, collaborative filtering and dimensionality reduction

Course outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	To learn the basic concepts and terminology in big data analytics	Familiarity
CO-2	To learn about the map reduce and the new software stack	Familiarity
CO-3	To learn about the mining of data streams, estimating moments and windowing, link analysis: page rank and efficient computation of page rank	Assessment
CO-4	To learn concepts associated with frequent item sets from big data and counting frequent items from stream	Assessment
CO-5	To learn about clustering for big data and mining of social network graph	Assessment
CO-6	To learn about recommendation systems, collaborative filtering and dimensionality reduction	Usage

Course Contents:

Unit	Contents	Lectures required
1	Introduction to Big Data: Big data time line, Why this topic is relevant now? Is big data fad? Where using big data makes a difference? Introduction to statistical modelling and machine learning, Ordinary data processing versus big data processing: Challenges and opportunities	**5**
2	Map Reduce and the New Software Stack: Distributed File Systems, Map Reduce, Algorithms Using Map Reduce, Complexity Theory for Map Reduce	**6**
3	Mining Data Streams: The Stream Data Model, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments and Windowing, Decaying Windows	**8**
4	Link Analysis: Page Rank and Efficient Computation of Page Rank, Topic-Sensitive Page Rank, Link Spam, Hubs and Authorities	**8**
5	Frequent Item sets from Big Data: The Market-Basket Model, Market Baskets and the A-Priori Algorithm, Handling Larger Datasets in Main Memory,	**8**

	Limited-Pass Algorithms, Counting Frequent Items in a Stream	
6	Clustering for Big Data: Introduction to Clustering Techniques, Hierarchical Clustering, Clustering in Non- Euclidean Spaces, Clustering for Streams and Parallelism	**7**
Total lectures		**42**

Suggested Text Book(s):

1. Anand Rajaraman and Jeffery David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012

Suggested Reference Book(s):

1. Jared Dean, Big Data, Data Mining and Machine Learning, Wiley Big data Series, 2014
2. Judith Hurwitz, Alan Nugent, Fern Halper and Marica Kaufman, Big Data for Dummies, Wiley Press, 2013

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106106142/
2. Link to topics related to course:
   - h. http://www.dbta.com/Columns/Big-Data-Notes/
   - i. https://people.cs.kuleuven.be/~joost.vennekens/DN/bigdata.pdf
   - j. https://www.tutorialride.com/big-databases/big-database-tutorial.htm

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Big Data)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	2	2	3	3	2	2	2	3	1	3	**2.4**
CO-5	3	3	3	2	3	3	2	2	3	3	2	2	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	2	**2.4**

Average	**3**	**3**	**2.8**	**2**	**2.7**	**2.8**	**2**	**2**	**2.3**	**3**	**1.2**	**2.7**	

Network Management

COURSE CODE: 18B1WCI845

COURSE CREDIT: 3

CORE/ELECTIVE: ELECTIVE

L-T-P: 3-0-0

Pre-requisites: Computer Networks, Cryptography and Network Security

Course Objectives:

1. Learn to develop applications to manage Networks.

Course Outcomes:

S.No.	Course Outcomes	Level of Attainment
CO-1	Become familiar with the Network Management Standards	Usage
CO-2	Understand the SNMP protocols	Usage
CO-3	Understand how large-scale Network Management Systems operate	Usage
CO-4	Understand how large-scale Network Management Systems are configured	Usage
CO-5	Advanced network Management Tools and Systems	Assessment
CO-6	Web Based Network Management Systems	Assessment

Course Contents:

Unit	Contents	Lectures required
1	**Data Communications and Network Management Overview** Review of Computer Network Technology, Basic Foundations of Network management, standards, models and languages	**10**
2	**Network Management Models** SNMP v1 Organisation and Information models, SNMP v1 Communication and functional Models, SNMP v2, SNMP v3, SNMP management RMON	**14**
3	**Network Design** Design of Data Communication Networks, Design of Tele Communication Networks, Design of Trasportation networks	**6**
4	**Broadband Network Management:** ATM Networks, Broadband Network Management: Access Networks, TMN	**6**
5	**Network Management Tools:** Network Management Tools systems and applications, Network Management applications	**4**
6	**Web Based Network Management:** Ubiquitous Web Based Network Management	**2**
Total lectures		**42**

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1

2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3.	T-3	35	2 Hours	Entire Syllabus
4.	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Suggested Textbook:

1. Mani Subramanian., Pearson Education, *Network Management Principals and Practices*

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Network Managemen)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	3	2	3	2	2	3	2	3	2	2	3	3	**2.5**
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	**2.6**
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	**2.4**
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	**2.5**
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	**2.4**
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	**2.6**
Average	**2.7**	**2.2**	**2.7**	**2.2**	**2.5**	**2.5**	**2.5**	**2.5**	**2.2**	**2.7**	**2.8**	**2.7**	

Graph Theory

COURSE CODE:

18B1WCI846

COURSE CREDIT: 3

CORE/ELECTIVE:

ELECTIVE L-T-P: 3-

0-0

Pre-requisites: None

Course Objectives:

1. To present a rigorous introduction to the fundamentals of Graph Theory and Graph algorithms.
2. To enable the students to model various applications from Computer Science and Engineering using Graphs.
3. To introduce the techniques to store, manipulate and answer queries about a graph using a computer.
4. To learn proof techniques and algorithms involving graphs.
5. To learn about open problems in graph theory

Course Outcomes:

S.No.	Course outcomes	Level of Attainment
CO-1	To learn the basic terminology and underlying principles of Graph theory	Familiarity
CO-2	To learn the Applications of Connectivity and Applications of Trees.	Assessment
CO-3	To learn the Applications of Matchings, Colourability, and Planarity.	Assessment
CO-4	Model real world problems using graph theory	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Scope, Basic concepts and terminology Representation: Adjacency Matrix, Incidence Matrix, Cycle Matrix, Cut-set Matrix, Path Matrix, etc	4
2	**Applications to Theoretical Computer Science:** Determining lower bounds, Adversary arguments, Problem reductions, NP-completeness, etc.	4
3	**Applications of Connectivity:** Reliable communication network design, Cycle detection, Searches, etc.	4
4	**Applications of Traversability:** Shortest paths, Optimal tours, TSP, etc. **Applications of Trees:** Spanning trees, Minimum cost constructions, Coding theory, Phylogeny construction, etc.	6
5	**Applications of Matchings/Partitioning:** Personnel assignment, Optimal assignment, Territory demarcation, etc.	6
6	**Applications of Coverings:** Geometric problems, etc.	6

	Applications of Colourability: Storage management, Timetable schedules, etc.	
7	**Applications of Planarity:** Planarity detection, PCB design, Facilities layout and floor plan design, Software testing, Defense strategies, etc.	6
8	**Applications of Digraphs:** Circuit theory and electrical network analysis, Transport networks, Job sequencing, Disk scheduling, Participant rankings in tournaments, Choice consistency, Project planning, etc. **Applications of Flows:** Max-flow min-cut, Feasible flows, Transportation problems, etc.	6
Total lectures		**42**

Suggested Text Book(s):

1. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, 1974.
2. Douglas B. West, Introduction to Graph Theory, PHI, second edition, 2001.

Suggested Reference Book(s):

1. Kenneth H. Rosen, Discrete Mathematics and its Applications, 6e, McGraw-Hill, 2007.
2. Thomas H Cormen, Charles E Leiserson, Ronald L. Rivest, and Cliff Stein,
3. Introduction to Algorithms, 2e, MIT Press, 2001.
4. Reinhard Diestel, Graph Theory, 3e, Springer-Verlag, 2005.
5. A Gibbons, Algorithmic Graph Theory, Cambridge University Press, 1985.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106108054/
2. Link to topics related to course:

i. https://swayam.gov.in/course/3795-graph-theory
ii. https://www.coursera.org/learn/graphs

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes (Graph Theory)	**PO-1**	**PO-2**	**PO-3**	**PO-4**	**PO-5**	**PO-6**	**PO-7**	**PO-8**	**PO-9**	**PO-10**	**PO-11**	**PO-12**	**Average**
CO-1	2	2	2	2	2	1	1	1	2	2	2	2	**1.8**
CO-2	2	3	3	3	3	1	1	1	2	2	1	2	**2**
CO-3	2	2	2	2	3	1	1	1	2	2	1	2	**1.8**
CO-4	2	3	3	3	2	1	1	1	2	3	2	2	**2.1**
Average	**2**	**2.5**	**2.5**	**2.5**	**2.5**	**1**	**1**	**1**	**2**	**2.3**	**1.5**	**2**	

Course description of Open Electives

Introduction to C++ Programming

COURSE CODE: 19B1WCI733

COURSE CREDITS: 2
ELECTIVE: OPEN ELECTIVE
L-T-P: 2-0-0

Pre-requisite: C Programming

Course Objectives:

The object oriented programming paradigm is one of the popular programming paradigms of today. Due to its characteristics object orientation has added new dimensions in the software development process. In this course concept of Object Oriented Programming (OOP) is introduced and for this purpose C++ programming language is being used. C++ a very powerful general purpose programming language, which supports object oriented programming paradigm. This course covers basics of C++ programming language which includes data types, variables, operators, and array and pointers. Also, object oriented features such as class and objects, inheritance, polymorphism are covered in this course. Finally exceptions handling, I/O operations and STL are explained.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Learn the concepts of an Object-Oriented Programming	Familiarity
CO-2	Learn to define class, objects, object members, class members, data members, constructors and destructors	Usage
CO-3	Learn the concepts of Inheritance and Polymorphism in C++	Usage
CO-4	Learn the advanced features of C++, such as file handling, templates, and exception handling	Usage

Course Contents:

S No.	Contents	Lectures required
1	Object Oriented Programming Structured vs. Object Oriented Programming, Object Oriented Programming Concepts, Benefits of Object oriented programming, Object Oriented Languages.	2
2	Objects and Classes classification, Defining Classes, Encapsulation, Instantiating Objects, Member Functions, Accessibility labels, Static Members.	5
3	Constructors and Destructors Purpose of Constructors, Default Constructor, Parameterized Constructors, Copy Constructor, Destructor, Memory Management	3

4	Inheritance Concept of Reusability, Types of Inheritance, Single and Multiple Inheritance, Multilevel Inheritance.	3
5	Operator Overloading Function and Operator Overloading, Overloading Unary and Binary Operators.	3
6	Polymorphism and Virtual Function Abstract Class, Function Overriding, Dynamic Binding, Pure Virtual Functions.	3
7	Streams and Files Stream Classes, Types of I/O, Formatting Outputs, File Pointers, Buffer.	3
8	Templates and STL Function and Class Templates, Use of Templates, Standard Template Library.	3
9	Exception Handling Exceptions in C++ Programs, Try and Catch Expressions, Exceptions with arguments.	3
	Total Lectures	**28**

Suggested Text Book(s):

1. Lafore R., Object oriented programming in C++, Waite Group
2. Stroustrap B., The C++ Programming Language, Addison Wesley
3. Bruce Eckel, Thinking in C++

Evaluation Scheme:

S No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 hour.	Syllabus covered upto Test-1
2	T-2	25	1.5 hours	Syllabus covered upto Test-2
3	T-3	35	2 hours	Syllabus covered upto Test-3
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programmed Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	1	2	2	1	2	2	1	1	1	3	3	**1.75**
CO-2	1	1	3	1	3	3	3	1	2	1	1	3	**1.92**
CO-3	2	2	1	2	1	1	2	3	1	3	2	1	**1.75**
CO-4	1	3	3	3	3	3	1	1	2	1	2	1	**2**
Average	**1.5**	**1.75**	**2.25**	**2**	**2**	**2.25**	**2**	**1.5**	**1.5**	**1.5**	**2**	**2**	

Introduction to C++ Programming Lab

COURSE CODE: 19B1WCI773
COURSE CREDITS: 1
CORE/ELECTIVE: OPEN ELECTIVE
L-T-P: 0-0-2

Pre-requisite: Introduction to C programming

Course Objectives:

The object oriented programming paradigm is one of the popular programming paradigms of today. Due to its characteristics object orientation has added new dimensions in the software development process. In this course concept of Object Oriented Programming (OOP) is introduced and for this purpose C++ programming language is being used. C++ a very powerful general purpose programming language, which supports object oriented programming paradigm. This course covers basics of C++ programming language which includes data types, variables, operators, and array and pointers. Also, object oriented features such as class and objects, inheritance, polymorphism are covered in this course. Finally exceptions handling, I/O operations and STL are explained.

Course Outcomes:

S No	Course Outcomes	Level of Attainment
CO-1	Learn the concepts of an Object-Oriented Programming	Familiarity
CO-2	Learn to define class, objects, object members, class members, data members, constructors and destructors	Usage
CO-3	Learn the concepts of Inheritance and Polymorphism in C++	Usage
CO-4	Learn the advanced features of C++, such as file handling, templates, and exception handling	Usage

List of Experiments

S No	Description	Hours
1	Design and implement a class using C++ that keeps the basic information about an account in a bank (the basic information must include account number and balance). The class should have a constructor that is used to create a new account object. The class should also have a function that transfers a requested amount from an account (identified by an account number) to another account.	**2**
2	Design and implement a class BCD using C++. The BCD class accepts an integer value (maximum five decimal digit input) using an input function of the class and which separates the digits and then stores them in a data member array of the class. For example, if input was 123 then it will be stored in data member array as num (0) = 3, num (1) = 2; and num (2) =1; similarly if input number was 2345 then it will be stored as num (0) = 5, num (1) = 4, num (2) = 3 and num (3) = 2. The class has another member function that calculates the reverse of integer value and prints it. For example, the input value 2345 will be reversed and printed as 5432.	**2**

3	Design and implement a class "circle". You may assume that the circle may be represented by the centre and the radius. The class should have functions to calculate the area of the circle and at least one overloaded operator function (you may overload << or >> or any other operator).	**2**
4	Define a class Employees having data members as: employee number, name, date of birth (dd-mm-yyyy), rank, and salary. When an employee is first recruited then all these are given values of 0. Upon confirmation, the actual values of these are entered for the employee. The employee's rank can be incremented by 1 and when this happens an employee gets an increment of 25%.	**2**
5	Students are registered in a University. When students are created then they are given default values (zeroes or blanks) for roll_number, department, year, and semester of study. At registration time, the values of these attributes of student are updated with the proper values. Students can be promoted and their departments can be changed.	**2**
6	Define two classes Distance1 and Distance2. Distance1 stores distance in miles and Distance2 in kmeters & meters. Write a program that reads values of the class objects and adds one object of Distance1 with the object of Distance2 class. The display should be in the format of miles or kmeters & meters depending on the type of object (Distance1 or Distance2) being used to invoke the function. (Hint: Make use of friend function).	**2**
7	Mid-Semester Practical Examination	**2**
8	Imagine a publishing company that markets both books and audio-cassette version of its works. Create a class Publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: Book, which adds a page count and Tape, which adds playing time in minutes. These classes should have getdata() function to get its data from the user and the putdata() function to display its data. Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in their data with getdata() and displaying the data with putdata().	**2**
9	Define a class Shape with two pure virtual functions as enterData() and displayArea(). Now, define a classes Rectangle and Circle that inherit the Shape class. Both the classes override and define the pure virtual functions. In the main() method, illustrate the concept of dynamic binding.	**2**
10	Define a class Directory with members: name and phone number. Use the class object to store each set of data into a text file "phone.txt". The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read the file and output the list in two columns, such as: John 23456 Denvar 9876 …… ……	**2**
11	Create a class named Student. Data members include the student's roll number, first name, last name, class and the program of study. Member functions include enter() and display() to enter and display the record of the	**2**

	student. In the main() method, write multiple objects of this class into the file stud.txt. There should be an option to search a student's record on the basis of his/her roll number and delete it from the file.	
12	A programmer wants to manipulate arrays. Two arrays are equal if (a) they have the same dimension, (b) are of the same size, and (c) contain identical values in their corresponding elements. Comparison is done using the operator '= =' which returns true or false. Also, arrays can be copied to one another using the operator '='. Define a template function to sort an array of elements of int type, float type, and string type.	**2**
13	Define an exception called "NoMatchException" that is thrown when a string is not equal to "India". Write a program that uses this exception.	**2**
14	End-Semester Practical Examination	**2**
	Total Lab hours	**28**

Evaluation Scheme:

S No	Exam	Coverage/Scope of Examination	Marks
1	Mid Term Test	Viva and Written Exam	20
2	End Term Test	Viva and Written Exam	20
3	Lab Records		15
4	Teacher Assessment	(Quality and quantity of experiment performed, learning laboratory skills)	30
5	Attendance and discipline in lab		15
6	Total		100

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	1	2	2	1	2	2	1	1	1	3	3	**1.75**
CO-2	1	1	3	1	3	3	3	1	2	1	1	3	**1.92**
CO-3	2	2	1	2	1	1	2	3	1	3	2	1	**1.75**
CO-4	1	3	3	3	3	3	1	1	2	1	2	1	**2**
Average	**1.5**	**1.75**	**2.25**	**2**	**2**	**2.25**	**2**	**1.5**	**1.5**	**1.5**	**2**	**2**	

Object Oriented Technologies using Java

COURSE CODE: 19B1WCI734

COURSE CREDITS: 2

ELECTIVE: OPEN ELECTIVE

: 2-0-0

Pre-requisite: Introduction to C Programming

Course Objectives:

Today almost every branch of computer science is feeling presence of object- orientation. Object oriented technology is successfully incorporated in various fields of computer science. Since its arrival on the scene in 1995, the Java has been accepted as one of the primary programming language. This course is designed to give you exposure to basic concepts of object-oriented technology. This course will help in learning to write programs in Java using object-oriented paradigm. Approach in this course is to take Java as a language that is used as a primary tool in many different areas of programming work.

Course Outcomes:

S No	Course Outcomes	Level of Attainment
CO1	Understanding the features and concepts of Object-Oriented Programming using Java	Familiarity
CO2	Defining classes, objects, constructors, methods in Java	Usage
CO3	Inheritance, Interfaces, and Polymorphism	Usage
CO4	Packages, Exception Handling, and Multithreading	Usage

Course Contents:

S No	Topics	Lectures required
1	Object-Oriented Concepts: Classes and Objects, Abstraction and Encapsulation, Inheritance, and Polymorphism.	2
2	Java Language Basics Introduction To Java, Basic Features, Java Virtual Machine Concepts, A Simple Java Program, Primitive Data Type And Variables, Java Keywords, Integer and Floating Point Data Type, Character and Boolean Types, Declaring and Initialization Variables, Java Operators.	3
3	Expressions, Statements and Arrays Expressions, Statements, Control Statements, Selection Statements, Iterative Statements, Jump Statements, Arrays.	2
4	Class and Objects Class Fundamentals, Creating objects, Assigning object reference variables, Introducing Methods, Static methods,	5
	Constructors, Overloading constructors, This Keyword, Using Objects as Parameters, Argument passing, Returning objects, Method Overloading, Garbage Collection, The finalize () Method.	

5	Strings and Characters Fundamentals of Characters and Strings, The String Class, String Operations, Data Conversion using valueOf() Methods, String Buffer Class and Methods.	2
6	Inheritance and Polymorphism Inheritance Basics, Access Control, Multilevel Inheritance, Method Overriding, Abstract Classes, Polymorphism, Final Keyword.	4
7	Packages and Interfaces Package, Defining Package, CLASSPATH, Package naming, Accessibility of Packages, Using Package Members, Interfaces, Implementing Interfaces, Interface and Abstract Classes, Extends and Implements Together.	3
8	Exceptions Handling Exception, Handling of Exception, Using try-catch, Catching Multiple Exceptions, Using finally clause, Types of Exceptions, Throwing Exceptions, Writing Exception Subclasses	3
9	Multithreaded Programming Multithreading: An Introduction, The Main Thread, Java Thread Model, Thread Priorities, Synchronization in Java, Inter-thread Communication.	4
	Total	**28**

Suggested Text Book(s):

1. Java 2: The Complete Reference, Fifth Edition -- by Herbert Schildt
2. Bruce Eckel, Thinking in Java

Evaluation Scheme:

S No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 hour.	Syllabus covered upto Test-1
2	T-2	25	1.5 hours	Syllabus covered upto Test-2
3	T-3	35	2 hours	Syllabus covered upto Test-3
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) -10 Attendance - 5

Course Outcomes (COs) contribution to the Programmed Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	1	2	2	1	2	2	1	1	1	3	3	**1.75**
CO-2	1	1	3	1	3	3	3	1	2	1	1	3	**1.92**
CO-3	2	2	1	2	1	1	2	3	1	3	2	1	**1.75**
CO-4	1	3	3	3	3	3	1	1	2	1	2	1	**2**
Average	**1.5**	**1.75**	**2.25**	**2**	**2**	**2.25**	**2**	**1.5**	**1.5**	**1.5**	**2**	**2**	

Object Oriented Technologies using Java Lab

COURSE CODE: 19B1WCI774
COURSE CREDITS: 1
CORE/ELECTIVE: OPEN ELECTIVE
L-T-P: 0-0-2
Pre-requisite: Introduction to C programming

Course Objectives:

Today almost every branch of computer science is feeling presence of object- orientation. Object oriented technology is successfully incorporated in various fields of computer science. Since its arrival on the scene in 1995, the Java has been accepted as one of the primary programming language. This course is designed to give you exposure to basic concepts of object-oriented technology. This course will help in learning to write programs in Java using object-oriented paradigm. Approach in this course is to take Java as a language that is used as a primary tool in many different areas of programming work.

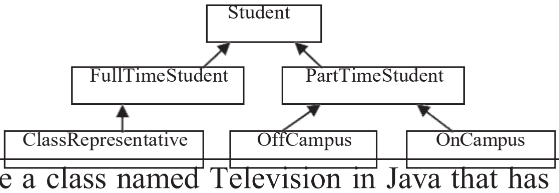
Course Outcomes:

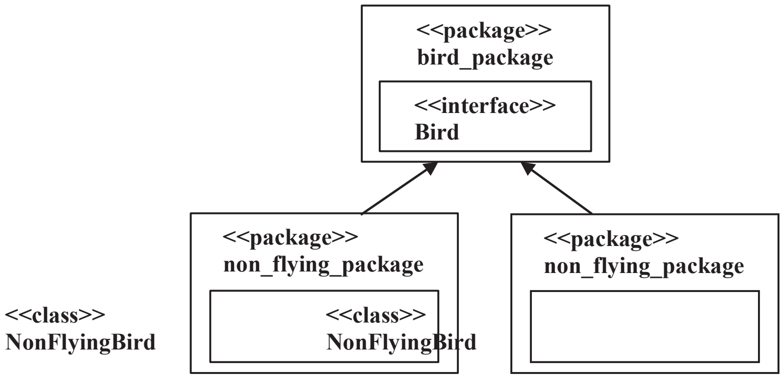
S No	Course Outcomes	Level of Attainment
CO1	Understanding the features and concepts of Object-Oriented Programming using Java	Familiarity
CO2	Defining classes, objects, constructors, methods in Java	Usage
CO3	Inheritance, Interfaces, and Polymorphism	Usage
CO4	Packages, Exception Handling, and Multithreading	Usage

List of Experiments

S No	Description	Hours
1	i. Write a Java program to check if a number is a palindrome in Java. ii. Write a Java program to print following structure. * *** ***** *** *	**2**
2	Write a program in Java to print the Fibonacci series up to a given number? Write both iterative and recursive versions.	**2**
3	Define a class Queue in Java with methods to insert and delete the data (int type) from the queue. The queue must be implemented using array.	**2**
4	Define a class LinkedList in Java with members data of integer type and a reference next of LinkedList type. Define methods to insert and display the elements of the linked list.	**2**

Approved in Academic Council held on 28 July 2021

5	Implement a singleton class. A class whose number of instances that can be instantiated is limited to one is called a singleton class. (Hint: make use of static members).	**2**
6	Write a Java program that accepts two strings S1 and S2 as input. The program should check if either of the input strings is a substring of the other and give appropriate message as output.	**2**
7	Mid-Semester Practical Examination	**2**
8	Design and implement the following class hierarchy using Java.  Student FullTimeStudent PartTimeStudent ClassRepresentative OffCampus OnCampus	**2**
9	Define a class named Television in Java that has data members to hold the model number of a television, the screen size in inches, and the price. The class has methods to enter and display the details of the television. The class throws an exception "TelevisionException under the following conditions: • If more than 4 digits are entered for the model number • If the screen size is smaller than 12 inches or greater than 70 inches. • If the price is negative or greater than Rs 50,000.	**2**
10	You may include additional data members in the classes. The class Programme is defined inside the package pack1.pack2, whereas classes FullTime and PartTime are defined inside the package pack1. You should include at least one constructor in each class. All the classes should have two methods as getProgrammeInfo() and printProgrammeInfo() to enter and display all the information of that object, respectively.	**2**
11	Define a class *First* inside a package pack1 with an integer data member *i* and methods to enter and display the value of *i*. Define another class inside a package pack2, which is a sub package of pack1. The class *Second* has a float type data member *f* and methods to enter and display the value of *f*. Now, import both of the above classes and instantiate their objects in the main method. Thereafter, call the methods of both the classes.	**2**
12	Realize the following class diagram using Java.	**2**

	 <<package>> bird_package <<interface>> Bird <<package>> non_flying_package <<class>> NonFlyingBird <<package>> non_flying_package <<class>> NonFlyingBird	
13	Write a program for generating two threads using Runnable interface. One thread is used for printing the numbers (1 to 26) and the other for printing alphabets (A to Z). Which of the two threads will be executed first (assuming that both of the threads are set to the same priority)?	**2**
14	End-Semester Practical Examination	**2**
	Total Lab hours	**28**

Evaluation Scheme:

S No	Exam	Coverage/Scope of Examination	Marks
1	Mid Term Test	Viva and Written Exam	20
2	End Term Test	Viva and Written Exam	20
3	Lab Records		15
4	Teacher Assessment	(Quality and quantity of experiment performed, learning laboratory skills)	30
5	Attendance and discipline in lab		15
6	Total		100

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	2	1	2	2	1	2	2	1	1	1	3	3	**1.75**
CO-2	1	1	3	1	3	3	3	1	2	1	1	3	**1.92**
CO-3	2	2	1	2	1	1	2	3	1	3	2	1	**1.75**
CO-4	1	3	3	3	3	3	1	1	2	1	2	1	**2**
Average	**1.5**	**1.75**	**2.25**	**2**	**2**	**2.25**	**2**	**1.5**	**1.5**	**1.5**	**2**	**2**	

Software Testing Methodologies

COURSE CODE: 19B1WCI735
COURSE CREDITS: 2
CORE/ELECTIVE: OPEN ELECTIVE
: 2-0-0

Pre-requisites: C/C++, Python, Eclipse, Netbeans

Course Objectives:

1. Employ correct testing terminology throughout the testing process
2. Execute specific software tests with well-defined objectives and targets.
3. Modelling techniques: UML: FSM and State charts, combinatorial design; and others.
4. Apply various testing techniques, including domain, code, fault, usage and model-based.
5. Perform a complete testing process, taking into account practical considerations**.**

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Investigate the reason for bugs and analyze the principles in software testing to prevent and remove bugs.	Familiarity
CO-2	Implement various test processes for quality improvement.	Assessment
CO-3	Design test planning and manage the test process.	Assessment
CO-4	Apply the software testing techniques in commercial environment.	Assessment
CO-5	Design test adequacy assessment and enhancement criteria.	Assessment
CO-6	Use practical knowledge of a variety of ways to test software and an understanding of some of the tradeoffs between testing techniques.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Error, Fault, Failure, Test automation and the importance of testing, Developer and tester as two roles, Principles of Testing, ETVX Model, Testing Maturity Model, V-Model, Software quality, Testing and debugging (preparing, Constructing, Executing, Specifying, Assessing a test plan), test Generation Strategies, Types of testing and Classifiers C1, C2, C3, C4, and C5, Static testing **Preliminaries mathematical:** Predicates and Boolean Expressions, Control Flow Graph, Program Dependence Graph, Strings languages and regular expressions,	**4**

2	**Test Generation:** a) **From Domain Partitioning:** The test selection problem, Equivalence partitioning, Boundary value analysis, Category-partition method, Cause-effect graphing. b) **From Finite State Models**: Finite State machines, Conformance testing, A Fault model, Characterization Set, The w-Method, The partial W-methos. c) **From Combinatorial design**: Combinatorial designs, A combinatorial test design process, Fault model, Latin Squares, Mutually orthogonal Latin squares, Pairwise designs: binary factors, Pairwise design: multi-valued factors, Orthogonal Arrays.	8
3	**Test Adequacy Assessment and Enhancement:** a) **Using Control flow:** Test adequacy basics, adequacy criteria based on control flow – Statement coverage, Decision coverage, condition coverage, MCC, LCSAJ, basis path coverage, b) **Using data Flow:** Definitions, C-use, p-use, Data flow graphs, du-path, dc-path, c-use coverage, p-use coverage, All-use coverage, k-dr chain coverage. c) **Using Mutation:** Mutation and Mutants, Test Assessment using mutation, Mutation operators, Founding principles of mutation testing, Equivalent mutants, Fault detection using mutation, Types of mutants.	8
4	**Phases of testing I:** **Regression testing:** Regression test process, Regression test selection, Selecting regression tests, test selection using execution trace, test selection using dynamic slicing	4
5	**Phases of testing II:** Unit Testing, Integration Testing, System testing, Acceptance testing.	4
Total lectures		**28**

Suggested Text Book(s):

1. "Foundations of Software testing," 2nd edition by Aditya P mathur, Pearson 2013
2. "Practical Software testing," 8th edition by Ilene Burnstein, Springer 2010

Suggested Reference Book(s):

1. Paul C. Jorgensen, Software testing: a Craft's man approach, CRC Press
2. Srinivasan Desikan and G. Ramesh, Software Testing: Principles and Practices, Pearson Education
3. Glenford Myers, "The Art of Software Testing", John Wiley & Sons Inc., New York, 1979.

4. Cem Kaner, Jack Falk, Nguyen Quoc, “Testing Computer Software”, Second Edition, Van Nostrand Reinhold, New York, 1993.
5. Boris Beizer, “Software Testing Techniques”, Second Volume, Second Edition, Van Nostrand Reinhold, New York, 1990.
6. Roger S. Pressman, “Software Engineering – A Practitioner’s Approach”, Fifth Edition, McGraw- Hill International Edition, New Delhi, 2001.
7. Boris Beizer, “Black-Box Testing – Techniques for Functional Testing of Software and Systems”, John Wiley & Sons Inc., New York, 1995.
8. K.K. Aggarwal & Yogesh Singh, “Software Engineering”, New Age International Publishers, New Delhi, 2003.

Other useful resource(s):

1. Link to NPTEL course contents: https://nptel.ac.in/courses/106105150/
2. Link to topics related to course:
   i. https://www.guru99.com/software-testing.html
   ii. https://www.inf.ed.ac.uk/teaching/courses/st/2011-12/Resource-folder/

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment		Entire Semester	Assignment (2) - 1 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes (POs)

Course outcomes (Software Testing Fundamentals)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-6	3	3	3	2	3	3	2	2	2	3	1	3	**2.5**
Average	**3**	**3**	**3**	**2**	**2.7**	**2.8**	**2**	**2**	**2.5**	**3**	**1**	**3**	

Software Testing Methodologies Lab

COURSE CODE: **19B1WCI775**

COURSE CREDITS: 1

CORE/ELECTIVE: ELECTIVE

L-T-P: 0-0-2

~~**Pre-requisite:** Object Oriented Analysis and Design with UML, Software Engineering, Software~~ Metrics, Basics of Mathematics.

Course Objectives:

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. To learn how to planning a test project, design test cases and data, conduct testing operations, manage software problems and defects, generate a testing report.
4. To discuss various software testing issues and solutions in software unit test; integration, regression, and system testing.
5. To gain the techniques and skills on how to use modern software testing tools to support software testing projects.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Have an ability to apply software testing knowledge and engineering methods.	Familiarity
CO-2	Have an ability to design and conduct a software test process for a software testing project.	Usage
CO-3	Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.	Assessment
CO-4	Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.	Assessment
CO-5	Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects.	Usage
CO-6	Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems.	Usage
CO-7	Have an ability to use software testing methods and modern software testing tools for their testing projects.	Usage

List of Experiments:

S. No	Description	Hours
1	Revision of Java coding using Eclipse	**2**

2	Developing Black box test cases 1: a) Using Boundary Value Analysis	**2**
3	Developing Black box test cases 2: a) Using Equivalent Classes	**2**
4	Developing Black box test cases 3: a) Using Decision Tables	**2**
5	Developing WhiteBox test cases 1: a) Performing Path testing	**2**
6	Developing WhiteBox test cases 2: a) Performing orthogonal testing	**2**
7	Developing WhiteBox test cases 3: a) Performing Coverage Analysis	**2**
8	Mutation testing and developing Mutants	**2**
9	Regression testing and Developing Regression test Cases	**2**
10	Performing GUI testing for a designed application	**2**
11	Performing Load testing for a designed application	**4**
12	Getting familiar with Profiler and performing CPU, Memory analysis in real time	**4**
	Total Lab Hours	**28**

Suggested Books/Resources:

1. 1. A Practitioner's Guide to Software Test Design, Lee Copeland, 2003,
2. The Art of Software Testing, 2nd edition, Glenford Myers, et. el., 2004
3. Software Testing Techniques, 2nd edition, Boris Beizer, 1990
4. How to Break Software: A Practical Guide to Testing, James Whittaker, 2002.
5. Testing Object-Oriented Systems: Models, Patterns, and Tools, Robert V. Binder, 1999.
6. Software Testing and Quality Assurance: Theory and Practice Paperback – 2010

Evaluation Scheme:

1	Mid Sem. Evaluation	20 Marks
2	End Sem. Evaluation	20 Marks
3	Attendance	15 Marks
4	Lab Assessment	45 Marks
	Total	100 marks

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	Average
CO-1	3	2	3	2	2	3	3	3	2	3	2	2	**2.5**
CO-2	3	3	3	2	3	3	3	3	2	3	2	3	**2.8**
CO-3	3	3	3	2	2	3	3	3	3	3	2	2	**2.7**
CO-4	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-5	3	3	3	2	2	2	3	3	3	3	2	2	**2.6**
CO-6	3	3	3	3	3	3	3	2	2	3	3	3	**2.8**
CO-7	3	3	3	3	3	3	2	2	3	3	3	3	**2.8**
Average	**3**	**2.9**	**3**	**2.4**	**2.6**	**2.9**	**2.9**	**2.6**	**2.4**	**3**	**2.4**	**2.6**	

Approved in Academic Council held on 28 July 2021

Software Defined Network

COURSE CODE: 19B1WCI739

COURSE CREDITS: 3

CORE/ELECTIVE: OPEN ELECTIVE

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

- To understand the fundamentals of software defined networks.
- To understand the separation of the data plane and the control plane.
- To study about the SDN Programming.
- To study about the various applications of SDN

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Analyze the evolution of software defined networks	Familiarity
CO-2	Express the various components of SDN and their uses	Assessment
CO-3	Explain the use of SDN in the current networking scenario	Assessment
CO-4	Design and develop various applications of SDN	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Basic Packet-Switching Terminology, Historical Background, The Modern Data Center, Traditional Switch Architecture, Autonomous and Dynamic Forwarding Tables. Open Source and Technological Shifts	7
2	**Why SDN?:** Evolution of Switches and Control Planes, Cost, SDN Implications for Research and Innovation, Data Center Innovation, Data Center Needs	7
3	**The Genesis of SDN :**The Evolution of Networking Technology, Forerunners of SDN, Software Defined Networking is Born., Sustaining SDN Interoperability, Open Source Contributions. Legacy Mechanisms Evolve Toward SDN, Network Virtualization	7
4	**How SDN Works:** Fundamental Characteristics of SDN, SDN Operation, SDN Devices, SDN Controller, SDN Applications, Alternate SDN Methods.	7
5	**The Open Flow Specification:** Open Flow Overview, Open Flow 1.0 and Open Flow Basics, Open Flow Limitations.	7
6	**Alternative Definitions of SDN and SDN in Other Environments:** SDN via APIs, SDN via Hypervisor-Based Overlays, SDN in Data Center, Wide Area Networks, Campus Networks, Hospitality Networks, Mobile Networks	7
Total lectures		**42**

Approved in Academic Council held on 28 July 2021

Suggested Text Book(s):

- Paul Goransson and Chuck Black, Software Defined Networks: A Comprehensive Approach, First Edition, Morgan Kaufmann, 2014.
- Thomas D. Nadeau, Ken Gray, SDN: Software Defined Networks, O'Reilly Media, 2013.

Suggested Reference Book(s):

- Siamak Azodolmolky, Software Defined Networking with Open Flow, Packet Publishing, 2013.
- Vivek Tiwari, SDN and Open Flow for Beginnersll, Amazon Digital Services, Inc., 2013.
- Fei Hu, Editor, Network Innovation through Open Flow and SDN: Principles and Design, CRC Press, 2014.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programmed Outcomes (POs)

Course Outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	1	2	2	2	3	1	1	1	1	2	2	3	**1.75**
CO-2	3	3	2	2	2	2	2	1	2	2	1	2	**2**
CO-3	3	1	2	1	2	2	2	2	2	3	1	1	**1.83**
CO-4	3	3	2	2	2	2	1	2	1	2	2	1	**1.91**
Average	**2.5**	**2.25**	**2**	**1.75**	**2.25**	**1.75**	**1.5**	**1.5**	**1.5**	**2.25**	**1.5**	**1.75**	

Introduction to Statistical Learning

COURSE CODE: **19B1WCI740**

COURSE CREDITS: 3

CORE/ELECTIVE: OPEN ELECTIVE

L-T-P: 3-0-0

Pre-requisite: Basics of programming and mathematics.

Course Objectives:

- To understand the fundamentals of statistical learning and, analyzing regression and classification.
- To understand and analyze pre-processing data and model selection.
- To understand and analyze non-linear and tree based regression.
- To understand and analyze Support Vector Classifier and Machines, and unsupervised learning.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understanding statistical learning and, analyzing regression and classification.	Familiarity and usage
CO-2	Analyzing resampling, model selection and regularization.	Usage
CO-3	Analyzing non-linear and tree based regression.	Usage
CO-4	Analyzing Support Vector Machines and unsupervised learning	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to statistical learning:** Prediction accuracy, Model interpretability, Supervised and Unsupervised learning, Assessing model accuracy	**6**
2	**Regression and Classification:** Regression- Linear, MultiLinear and kNN, Classification-Logistic regression, Linear and quadratic discriminant analysis, and kNN	**8**
3	**Resampling, Model Selection and Regularization:** Cross- validation, the bootstrap, Subset selection, Shrinkage methods, Dimension reduction methods, considerations in high dimension	**8**
4	**Non-linear Regression:** Polynomial regression, Step functions, Regression splines, Smoothing splines, Local regression, Generalized additive methods, **Tree Based Regression:** the basics of decision trees, Bagging, Random forest and Boosting,	**10**
5	**Support Vector Machines:** Maximal margin classifier, Support Vector Classifier and Support Vector Machines, **Unsupervised Learning:** The challenges, Principal Component Analysis, Clustering methods	**10**
	Total lectures	**42**

Suggested Text Book(s):

- Gareth James, Daneila Witten, Trevor Hastie and Robert Tibshirani, "An Introduction to Statistical Learning with Applications in R", Springer, 8th Printing, 2017.

Suggested Reference Book(s):

- Hastie, Tibshirani and Friedman, "The elements of statistical learning", 2nd edition, Springer, 2009.
- Sanjeev Kulkarni and Gilbert Harman, " An elementary introduction to statistical learning theory, Wiley, 2011.
- Fernandes de Mello, Rodrigo et. al., " Machine Learning – A practical approach on the statistical learning theory", Springer, 2018.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programmed Outcomes (POs)

Course Outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	3	3	3	1	1	2	1	3	2	**2.33**
CO-2	3	3	3	3	3	3	2	1	2	1	3	2	**2.42**
CO-3	3	3	3	3	3	3	2	1	2	1	3	2	**1.42**
CO-4	3	3	3	3	3	3	1	1	2	1	3	2	**2.33**
Average	**3**	**3**	**3**	**3**	**3**	**3**	**1.5**	**1**	**3**	**1**	**3**	**3**	

Principles of Distributed Database Systems

COURSE CODE: 19B1WCI838

COURSE CREDITS: 3

CORE/ELECTIVE: OPEN ELECTIVE

L-T-P: 3-0-0

~~**Pre-requisite:** Database management Systems~~

Course Objectives:

To introduce the fundamental concepts and design issues of managing large volume of shared data in distributed environment, and to provide insight into related research problems.

- To understand the fundamentals of Distributed Database Systems.
- To understand the various database designing processes.
- To study about the query processing and query optimization in a distributed environments.
- To study about the various applications of distributed databases.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Understand the fundamentals of Distributed Database Systems (DDBMS), including its design and architecture	Familiarity
CO-2	Understand the concept of data integration design process	Assessment
CO-3	Gain skill about query processing & optimization in distributes environment	Assessment
CO-4	Explain the use of Distributed database system in real life applications	Usage
CO-5	Implement applications/tools in order to utilize DDBMS applications.	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Distributed Data Processing, What is a Distributed Database System? Data Delivery Alternatives, Promises of DDBSs, Complications Introduced by Distribution, Design Issues, Distributed DBMS Architecture,	4
2	**Distributed Database Design:** Top-Down Design Process, Distribution Design Issues, Fragmentation, Allocation, Data Directory	7
3	**Database Integration:** Bottom-Up Design Methodology, Schema Matching, Schema Integration, Schema Mapping, Data Cleaning	7
4	**Overview of Query Processing:** Query Processing Problem, Objectives of Query Processing, Complexity of Relational Algebra Operations, Characterization of Query Processors, Layers of Query	6

	Processing, Query Decomposition, Localization of Distributed data	
5	**Optimization of Distributed Queries:** Query Optimization, Centralized Query Optimization, Join Ordering in Distributed Queries, Distributed Query Optimization	6
6	**Multidatabase Query Processing:** Issues in Multidatabase Query Processing, Multidatabase Query Processing Architecture, Query Rewriting Using Views, Query Optimization and Execution, Query Translation and Execution	6
7	**Dataspace System:** Introduction, Pay-as-you-go data integration, Design issues, Data modeling, Query processing and answering, Entity resolution, Users feedback in Dataspace System, Indexing dataspaces, Source Discovery	6
	Total lectures	**42**

Suggested Text Book(s):

- Principles of Distributed Database Systems, M. Tamer Ozsu, Patrick Valduriez , Springer, 2011, ISBN 978-1-4419-8833-1.
- Distributed Database Management Systems- A Practical Approach, Saeed K Rahimi, Frank S Haug, Wiley Publication, 2010, ISBN 978-0-470-40745-5

Suggested Reference Book(s):

1. Distributed Database Mangement Systems, Rahimi & Haug, Wiley
2. Distributed Database Systems, Chanda Ray, Pearson Publication
3. Additional research papers from the literature may be assigned as reading material.

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) – 10 Quizzes (2) - 10 Attendance – 5

Course Outcomes (COs) contribution to the Programmed Outcomes (Pos)

Course outcomes (Software Defined Network)	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**
CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**

Approved in Academic Council held on 28 July 2021

Average	**3**	**3**	**3**	**2**	**2.6**	**2.8**	**2**	**2**	**2.6**	**3**	**1**	**3**	

Foundations of Block Chain

Course Code: 19B1WCI839

Course Credit: 3-0-0

Course Pre-requisites (if any): Some programming experience (at least one year suggested). Potential students will need to pass a brief programming assessment for enrollment in the course.

Course Objectives:

S No	Course Outcomes (Block Chain Technologies) 18B1WCI836	Level of Attainment
CO-1	Explain how bitcoin works, from when a transaction is created to when it is considered part of the blockchain	Familiarity
CO-2	Thoroughly explain private and public keys as well as addresses and how exactly they are constructed and used	Familiarity
CO-3	Expose students to the different kinds of forking and explain the Bitcoin's network mechanisms for maintaining and upgrading	Assessment
CO-4	Decompose a blockchain system's fundamental components, how they fit together and examine a modular blockchain system in more detail	Assessment
CO-5	Detailed understanding of naïve Attacks and Trustless Networks of block chain	Assessment
CO-6	Provide a thorough understanding of smart contracts, their technical capabilities, practical applications, limitations and security constraints they operate within	Usage
CO-7	Explain to students both fundamental and implied differences between Ethereum and Bitcoin protocol by covering historical, conceptual and architectural distinctions	Assessment

Detailed Course Contents:

Module	Contents	Lectures required
Module-1	**The story of a transaction** a) From Transactions to Blocks b) Blocks and Distributed Consensus c) Basic interaction with a Bitcoin node	**2**
Module-2	**Keys and Addresses** a) Basic cryptography b) From private keys to addresses	**2**
Module-3	**The Bitcoin Script language** a) Introduction to the Bitcoin Script language b) Script writing and execution c) Advanced scripting	**8**
	d) Tools and libraries to access Bitcoin's API and scripting capabilities **Blockchain deployment** a) Mining and forking	

	b) Upgrading the network c) Related BIPs d) Segregated Witness (SegWit)	
Module-4	**Blockchain architectures** a) Abstract Architecture b) Ways to dive deeper c) Introduction to major blockchain platforms	**6**
Module-5	**Comparing Bitcoin and Ethereum** a) Historical comparison b) Conceptual distinction between a payment system and a decentralized applications platform *c)* Differences in their architectures from *security-first aspect to a rich feature set* d) Future roadmap for them, following their own paths with probable interconnections	**8**
Module-6	**Development environment** a) Multitude of clients in Ethereum b) Production and test networks in Ethereum *c) Public, private and development deployments* **Contract code walk-through** a) Demonstration of smart contract b) Introduction to Solidity c) Contract lifecycle	**8**
Module-7	**Solidity in depth** a) Building blocks b) Popular contracts already in deployment **Considerations for production deployment** a) Quality of decentralized applications b) Code patterns *c) Security* d) Other smart contract platforms e) Discussion of future prospects	**8**
	TOTAL LECTURES	**42**

Suggested Text Book(s):

- Mastering Bitcoin, Andreas Antonopoulos, O'Reilly Publishing, 2014, ISBN: 978-0691171692.
- Bitcoin: A Peer-to-Peer Electronic Cash System, Satoshi Nakamoto, Online2009 https://bitcoin.org/bitcoin.pdf
- Vitalik Buterin, Ethereum White Paper, Online, 2017, https://github.com/ethereum/ wiki/wiki/ White- Paper

Suggested Reference Book(s):

- Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction, A. Narayanan, J.
- Bonneau, E. Felten, A. Miller, S. Goldfeder, Princeton University Press, 2016, ISBN: 978-0691171692
- The Science of the Blockchain, Roger Wattenhofer, CreateSpace Independent Publishing Platform, 2016, ISBN: 978-1522751830
- Ethereum Programming, Alex Leverington, Packt Publishing Limited, 2017, ISBN: 978- 1786463715

Other useful resource(s):

- Bitcoin Protocol Specifications (https://en.bitcoin.it/wiki/Protocol_specification)
- Bitcoin transaction Scripting (https://en.bitcoin.it/wiki/Script)
- Majority is not Enough: Bitcoin Mining is Vulnerable (http://arxiv.org/abs/1311.0243)
- Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin (http://eprint.iacr.org/2012/248.pdf)
- Ethereum documentation (http://www.ethdocs.org/en/latest)
- Solidity documentation ((https://solidity.readthedocs.io/en/develop))
- https://sites.google.com/site/blockchaintutorial/blockchain-course-content

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) – 10 Quizzes (2) - 10 Attendance – 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Weightage
CO-1	3	2	3	2	2	3	2	3	2	2	3	3	83
CO-2	3	2	3	2	2	3	3	2	2	3	3	3	86
CO-3	2	2	2	2	2	3	3	3	2	2	3	3	81
CO-4	3	2	3	2	3	2	2	3	3	3	2	2	83
CO-5	3	2	2	2	3	2	2	2	2	3	3	3	81
CO-6	2	3	3	3	3	2	3	2	2	3	3	2	86
CO-7	2	2	2	2	2	3	3	3	2	2	3	3	81
Weightage	86	71	86	71	81	86	86	86	71	86	95	90	

Computational Biology

COURSE CODE: 19B1WCI840

COURSE CREDITS: 3

~~**ELECTIVE:** OPEN ELECTIVE~~

L-T-P: 3-0-0

Pre-requisite: None

Course Objectives:

- To understand the fundamentals of Computational Molecular Biology.
- To understand the computational methods for biological sequence analysis.
- To study about the dynamic programming for sequence alignments, methods for genomics and structural bioinformatics.
- To study about the various applications of computer science techniques in computational biology.

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Fundamentals of Computational Molecular Biology	Familiarity
CO-2	Express the use of computational methods for biological sequence analysis	Assessment
CO-3	Explain the use of sequence alignments, methods for genomics and structural bioinformatics	Assessment
CO-4	Applications of computer science techniques in computational biology	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction:** Basic knowledge of molecular biology including DNA, RNA, Proteins and Genetic code systems. Biological sequences types and data. Computers in biology, medicine and healthcare.	**5**
2	**Biological Databases:** Evolution of biological databases. Collection, storage and analysis of biological data through biological databases. Sequence formats and storage, Access databases, limitations of existing databases.	**4**
3	**Sequence Alignment and Analysis:** Local alignment, Global alignment, Scoring matrices - PAM, BLOSUM, Gaps and penalties, Dot plots. Dynamic programming approach: Needleman and Wunsch Algorithm, Smith and Waterman Algorithm, Hidden Markov Model: Viterbi Algorithm. Heuristic approach: BLAST, FASTA. Building Profiles, Profile based functional identification.	**7**
4	**Computational Genomics:** Introduction to Next Generation Sequencing technologies. Annotations through Functional and comparative genomics approaches, Human genome project. GWAS, ENCODE, HUGO projects. Personalized and precision medicines. **Software tools in Biology:** Visualization tools including Artemis and Vista for genome comparison. Applications of Grid computing in	**10**

	Biology. Compute Clusters and nature of problems dealt with them.	
5	**Algorithms in Computational Biology:** Whole Genome Assembly and challenges, Fragment assembly algorithms, semi- global alignment, exon chaining and spliced algorithms, brute force, median string and other motif finding algorithms. Recent algorithms for MSA.	**6**
6	**Chemoinformatics:** Introduction, representing 2D & 3D structures, 2D chemical database applications & molecular descriptors and their classifications, database searching and applications in computer aided drug design.	**5**
7	**Quantitative structure activity relationship (QSAR)**: QSAR methodology, biological and physicochemical parameters, feature selection (PLS, PCA, MLR, etc.), model building and validation, QSAR applications in drug design, Quantitative structure-property relationships (QSPR), CoMFA, 3D and nD-QSAR methods.	**5**
	Total lectures	**42**

Suggested Text and Reference Book(s):

1. Dan E. Krane and Michael L. Raymer, 'Fundamental concepts of Bioinformatics', Pearson Education (low Priced Edition).
2. Claverie & Notredame, 'Bioinformatics - A Beginners Guide', Wiley- Dreamtech India Pvt Ltd.
3. J. Pevnezer, 'Bioinformatics and functional genomics', John Wiley.
4. David Mount, Bioinformatics: sequence and Genome Analysis, Latest Edition. Cold Spring Harbour Labortory Press.
5. Discovering Genomics, Proteomics and bioinformatics, 2/E by A. Malcolm Campbell and Laurie J.Heyer, Publisher:Benjamin Cummings.
6. ICRF handbook of genome analysis, by NK Spurr, BD Young, SP Bryant. Volumes I & II. - Blackwell science publishers.
7. Isaac S. Kohane, Alvin T. Kho, Atul J. Butte Microarrays for an Integrative Genomics, MIT Press.
8. Philip E. Bourne, Jenny Gu, Structural Bioinformatics, Wiley Blackwell Publisher.

Evaluation Scheme:

S No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment	25	Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programmed Outcomes (POs)

Course Outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	1	2	2	2	3	1	1	1	1	2	2	3	**1.75**
CO-2	3	3	2	2	2	2	2	1	2	2	1	2	**2**

CO-3	3	1	2	1	2	2	2	2	2	3	1	1	**1.83**
CO-4	3	3	2	2	2	2	1	2	1	2	2	1	**1.91**
Average	**2.5**	**2.25**	**2**	**1.75**	**2.25**	**1.75**	**1.5**	**1.5**	**1.5**	**2.25**	**1.5**	**1.75**	

Multimedia Systems and Applications

COURSE CODE: 19B1WCI843

COURSE CREDITS: 3

CORE/ELECTIVE: OPEN ELECTIVE

: 3-0-0

Pre-requisites: Computer Networks, Operating Systems, Data Structure, Java Programming, Linear Algebra, Web Technology (Those who are simultaneously registered in some of these courses can also take this course.)

Course Objectives:

- fundamentals of multimedia systems
- basic concepts related to multimedia communication.
- the evolution, latest trends, and state-of-the-art in multimedia technology, standards, and applications

  Overall, the aim of the course would be to cover all aspects related to multimedia systems and applications ranging from basic concepts and fundamentals to more advanced discussions on the state-of-the-art in this field..

Course Outcomes:

S. No.	Course Outcomes	Level of Attainment
CO-1	Demonstrate mastery of scripting in a multimedia development environment.	Familiarity
CO-2	Create multi-user multimedia applications.	Assessment
CO-3	Create an image/audio/video multimedia application.	Assessment
CO-4	Apply image-processing algorithms to multimedia content within a scripting environment.	Assessment
CO-5	Apply current standards and guidelines for multimedia application development and delivery	Usage

Course Contents:

Unit	Contents	Lectures required
1	**Introduction to Multimedia System and Processing** Introduction to Multimedia, Application Areas, Interdisciplinary Aspects of Multimedia, Multimedia Data representations, Multimedia Data Encoding (Audio, Image, Video and Animation),	**6**
2	**Multimedia Compression Basics** Concept of data compression in multimedia field, lossless techniques (Huffman Coding, Arithmetic and Lempel-Ziv Coding, Other Coding Techniques) and lossy compression techniques (Transform Coding & K-L Transforms, Discrete Cosine Transforms, and BTC), Multi- Resolution Analysis, and Still Image Compression Standards (JBIG and JPEG), Color image processing.	**12**

3	**Audio and Video Processing:** Basics of digital audio, quantization and transmission of Audio. Audio compression, Audio MPEG, Video Coding Standards (MPEG video coding, MPEG4, 7, and beyond).	**6**
4	**Media Server & Networks :** Media Server Architecture, Storage Management, Services, Protocols, Layers, Requirements to Services and Protocols, Layers of the ISO-OSI Model, ATM Networks, Traditional network protocols and their support for Multimedia, Traditional transport protocols and their support for Multimedia, New protocols for transport of multimedia	**6**
5	**Quality of Service & Multimedia Operating System** Requirements and Constraints, Quality of Services Concept, Resource Management, Establishment Phase (QoS Translation, QoS Scaling, QoS Routing, Admission Control), Run-time Phase of Multimedia Call, Process Management, Inter process Communication and Synchronization, Memory Management, Device Management.	**6**
6	**Multimedia Applications:** Case studies	**6**
	Total lectures	**42**

Suggested Text Book(s):

- Ralf Steinmetz, Klara Nahrstedt. Multimedia Systems, Springer, Springer International Edition, (Textbook)
- Fundamentals of Multimedia: Ze-Nian Li & Mark S. Drew, Pearson Prentice Hall, 2004

Suggested Reference Book(s):

- John. F. Koegel Buford. Multimedia Systems. Pearson Education.
- Robert Reinhardt and Joey Lott. Flash MX Action Script Programming. Wiley.
- James E. Shuman. Multimedia in Action. Cengage Learning.
- Khalid Shayood, Data Compression.

Other useful resource(s):

- IEEE Transactions on Multimedia Systems.
- Multimedia Systems (ACM Press)
- ACM SIGMM conference

Evaluation Scheme:

S. No	Exam	Marks	Duration	Coverage / Scope of Examination
1	T-1	15	1 Hour.	Syllabus covered upto T-1
2	T-2	25	1.5 Hours	Syllabus covered upto T-2
3	T-3	35	2 Hours	Entire Syllabus
4	Teaching Assessment		Entire Semester	Assignment (2) - 10 Quizzes (2) - 10 Attendance - 5

Course Outcomes (COs) contribution to the Programme Outcomes(POs)

Course Outcomes	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PO-12	Average
CO-1	3	3	3	2	2	3	2	2	2	3	1	3	**2.4**
CO-2	3	3	3	2	3	2	3	2	2	3	1	3	**2.5**
CO-3	3	3	3	2	2	3	1	2	3	3	1	3	**2.4**

CO-4	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
CO-5	3	3	3	2	3	3	2	2	3	3	1	3	**2.6**
Average	**3**	**3**	**3**	**2**	**2.6**	**2.8**	**2**	**2**	**2.6**	**3**	**1**	**3**	