JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

Criteria	2 Teaching-learning and Evaluation
Key Indicator	2.6 Student Performance and Learning Outcomes
Metric	2.6.1 The institution has stated learning outcomes (generic and programme specific) / graduate attributes which are integrated into assessment process and widely publicized through the website and other documents

COs of All Courses – Department of Computer Science & Engineering and Information Technology (BTech-IT)

Semester: I

SNo	Course Codes	Course Name	Course Outcomes
1	10B11CI111	Introduction to Computer and Programming	CO1: Students will be acquainted with Data & Instruction representation, CO2: To gain knowledge of Architecture and functional operation of computer peripherals. CO3: To learn User-interface of DOS, Windows and UNIX. CO4: Ability to design, develop, test and document structured programs in C language.
2	10B17CI171	Computer Programming Lab	CO1: Understand the terminology used in computer programming CO2: Design programs involving decision structures, loops and functions. CO3: Understand the dynamics of memory by the use of pointers. CO4: Enhance programming skills through problem solving and code development of small-size software applications.
3	18B11CI111	Programming for Problem Solving Lab	CO1: Students will be acquainted with Data & Instruction representation, CO2: To gain knowledge of Architecture and functional operation of computer peripherals. CO3: To learn User-interface of DOS, Windows and UNIX. CO4: Ability to design, develop, test and document structured programs in C language.
4	18B11CI111	Programming for Problem Solving	CO1: Understand the terminology used in computer programming CO2: Design programs involving decision structures, loops and functions. CO3: Understand the dynamics of memory by the use of pointers. CO4: Enhance programming skills through problem solving and code development of small-size software applications.
5	10B17CI171	Computer Programming Lab	CO1: Understand the terminology used in computer programming CO2: Design programs involving decision structures, loops and functions. CO3: Understand the dynamics of memory by the use of pointers. CO4: Enhance programming skills through problem solving and code development of small-size software applications.

| 6 | 19B11CI111 | Programming for Problem Solving-II | CO-1 To formulate simple algorithms for arithmetic and logical problems.
CO-2 To translate the algorithms to programs (in C language).
CO-3 To test and execute the programs and correct syntax and logical errors.
CO-4 To implement conditional branching, iteration and recursion.
CO-5 To decompose a problem into functions and synthesize a complete program using divide and conquer approach.
CO-6 To use arrays, pointers and structures to formulate algorithms and programs.
CO-7 To apply programming to solve matrix addition and multiplication problems and searching and sorting problems.
CO-8 To apply programming to solve simple numerical method problems, namely rot finding of function, differentiation of function and simple integration. |
| 7 | 19B17CI171 | Programming for Problem Solving-II Lab | CO-1 Develop problem solving ability using programming.
CO- 2 To impart adequate knowledge on the need of programming languages and problem solving techniques.
CO-3 To develop a methodological way of problem solving.
CO-4 Analyze and construct effective algorithms.
CO-5 Employ good programming practices such as incremental development, data integrity checking and adherence to style guidelines.
CO-6 Learn a programming approach to solve problems. |

Semester: II

SNo	Course Codes	Course Name	Course Outcomes
1.	18B11CI211	Data Structures and Algorithms	CO-1 To gain knowledge on the notions of data structure, Abstract Data Type. CO-2 For a given algorithm student will able to analyze the algorithms to determine the time and computation complexity and justify the correctness. CO-3 For a given Search problem (Linear Search and Binary Search) student will able to implement it. 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. CO-5 Student will able to implement Graph search and traversal algorithms and determine the time and computation complexity. 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.
2.	18B11CI271	Data Structures and Algorithms Lab	CO-1 To gain knowledge on the notions of data structure, Abstract Data Type CO-2 To have hands on skills to evaluate different kinds of linked lists and their applications in day to day problem solving. CO-3 To have hands on skills to evaluate different kinds stacks and their applications and implementations in day to day problem solving CO-4 To have hands on skills to evaluate different kinds queues and their applications and implementations in simulations. CO-5 To acquire knowledge of various sorting algorithms CO-6 To learn Searching: Balanced tree, red-black tree, lower bounds for searching CO-7 To learn to code for operations on Tree or BST (Creation; Traversing like preorder,post-order and in-order; Searching element; finding height etc.) CO-8 Introduction to Heaps CO-9 To learn to code for operations on Graphs (Creation; entering info, printing output and deleting; traversal of BFS and DFS algorithm)
3.	10B11CI211	Data Structures	CO 1: Understand the notions of data structure, Abstract Data Type. CO 2: Explore asymptotic notations and role of algorithm complexity in computing CO 3: Evaluate different kinds of linked lists and their applications in day to day problem solving. CO 4: Evaluate different kinds stacks and their applications and implementations in day to day problem solving. CO 5: Evaluate different kinds queues and their applications and implementations in simulations.

			CO 6: To acquire knowledge of various sorting algorithms CO 7: To learn Searching: Balanced tree, red-black tree, lower bounds for searching CO 8: To gain understanding of Graph: representation and algorithms CO 9: To have exposure to hashing, and its various implementations of searching and hashing algorithms
4.	10B17CI271	Data Structures and Computer Programming Lab	CO-1: Develop problem solving ability using Programming. CO-2: Develop ability to design and analyze algorithms. CO-3: Introduce students to data abstraction and fundamental data structures. CO-4: Develop ability to design and evaluate Abstract Data Types and data structures. CO-5: Apply data structure concepts to various examples and real life applications
5.	14B21CI211	Basic Data Structures	CO1: To gain knowledge on the notions of data structure, Abstract Data Type. CO2: To have exposure to Big(O) notation and role of algorithm complexity in computing CO3: To have hands on skills to evaluate different kinds of linked lists and their applications in day to day problem solving. CO4: To have hands on skills to evaluate different kinds stacks and their applications and implementations in day to day problem solving. CO5: To have hands on skills to evaluate different kinds queues and their applications and implementations in simulations. CO6: To acquire knowledge of various sorting algorithms CO7: To learn Searching: Balanced tree, red-black tree, lower bounds for searching CO8: To gain understanding of Graph: representation and algorithms CO9: To have exposure to hashing, and its various implementations of searching and hashing algorithms.

Semester: III

SNo	Course Codes	Course Name	Course Outcomes
1.	10811CI311	Object Oriented Programming	CO1: Explain what constitutes an object-oriented approach to programming and identify potential benefits of Object-oriented programming over other approaches. CO2: Analyze and decompose problem specifications from Object Oriented Perspectives and represent the solution, using UML notation. CO3: Explain the benefits of object oriented design and the types of systems in which it is an appropriate methodology. CO4: Apply an object-oriented approach to developing applications of varying complexities. CO5: Augment a class definition using constructors, destructors, member functions, helper functions and custom input/output operators to add functionality to a programming solution CO6: Manage an object's resources using dynamic memory allocation and deal location to access data stored outside the object's memory CO7: Read from and write to files using objects from the standard input output library and custom file operators for future restoration CO8: Model specialization using single inheritance and abstract base classes to minimize code duplication CO9: Model polymorphic behavior using coercion, overloading, virtual functions and function templates to amplify reusability of code
2.	10811CI312	Database Systems	CO1: Explain the characteristics, architecture of database approach, describe the components, major functions of a database system and give examples of their use. CO2: Compare and contrast appropriate data models, including concepts in modeling notation and how they would be used. CO3: Demonstrate use of the relational algebra operations from mathematical set theory (union, intersection, difference, and Cartesian product) and the relational algebra operations developed specifically for relational databases (select (restrict), project, join, and division).

			CO4: Create a relational database schema in SQL, use SQL to create a non-procedural query, write a stored procedure that deals with parameters and has some control flow, to provide a given functionality. CO5: Using SQL to implement roles, privileges, access control and authorization policies CO6: Determine the functional dependency between two or more attributes, compute the closure of a set of attributes, evaluate a proposed decomposition, describe the properties of BCNF, PJNF, 5NF. CO7: Explain the use of integrating OO properties with relational modeling CO8: Give examples of the application of primary, secondary, and clustering indexes, explain the theory and application of internal and external hashing techniques. CO9: Understand a set of query processing strategies and select the optimal strategy. CO10: Understand transaction management, concurrency control techniques and data recovery methods. CO11: Explain the techniques used for data fragmentation, replication, evaluate simple strategies for executing a distributed query and explain how the two-phase commit
3.	10817CI307	Unix Programming Lab	CO1: You will be able to run various UNIX commands on a standard UNIX/LINUX Operating system. CO2: You will be able to run C programs on UNIX. CO3: You will be able to do shell programming on UNIX OS. CO4: You will be able to understand and handle UNIX system calls.
4.	10817CI371	Objected Oriented Programming Lab	CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors CO2: To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions. CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level CO4: Inheritance, Hierarchical Inheritance and Hybrid Inheritance. CO5: To learn the concept of Abstract classes and interfaces

			CO6: To learn the concepts of Operator overloading and conversion function CO7: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file. CO8: To learn the Exception Handling: try -- catch and finally block, making user-defined exceptions. CO9: To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.
5.	10817CI372	Database Systems Lab	CO1: Students get practical knowledge on designing and creating relational database systems. CO2: Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger, views and embedded SQL. CO3: Use of various software to design and build ER Diagrams, UML, Flow chart for related database systems. CO4: Students will be able to design and implement database applications on their own
6.	10828CI408	Multimedia Development Lab-I	CO1: To learn how to design and develop multimedia for real world e-learning. CO2: To practice software engineering in a learner-oriented project. CO3: To learn and apply cognitive principles of user interface design. CO4: To learn how to implement multimedia e-leaning in Macromedia Flash. CO5: Demonstrate mastery of scripting in a multimedia development environment. CO6: Create multi-user multimedia applications. CO7: Create a 3D multimedia application. CO8: Apply image-processing algorithms to multimedia content within a scripting environment. CO9: Apply current standards and guidelines for multimedia development and delivery. CO10: Create production quality multimedia applications. CO11: Use the basic tools used by an Android programmer. CO12: Configure an Android emulator and a

			hardware connection to an Android device. CO13: Describe the Android development lifecycle. CO14: Define XML and give examples of how it is used to express data. CO15: Write interactive programs on the Android. CO16: Demonstrate the use of activity lifecycles to control an app. CO17: Analyze a design's ability to support multiple screen resolutions and natural languages. CO18: Design dynamic UIs using fragments and the Android support library. CO19: Analyze a design's ability to interact with other apps on the device. CO20: Use layout hierarchies to produce reusable layouts .
7.	18B11CI315	Python Programming with Raspberry Pi	CO-1 Introducing Raspberry pi: what is programming, why raspberry pi, introducing the raspberry pi CO-2 Introducing Python: Scripting vs. a Programming Language, The Python Philosophy, Getting Started with Python, Python basics, Programming with Python CO-3 Graphical Programming: Graphical User Interface (GUI) Programming, Adding Controls, Creating a Web Browser, Adding Window Menus, the web bolt, Calculating the Distance of the Point from the Light Source. CO-4 Creating games: building a game, installing Pi game, creating a worlds, detecting collisions, adding sound, adding scenery, adding the finishing touches, realistic game physics, Calculating Reflecting Angles, Building the 3D Model, Speaking to Your Pi. CO-5 Networked Python: Understanding Hosts, Ports, and Sockets, Locating Computers with IP Addresses, Building a Chat Server, Tweeting to the World. CO-6 Multimedia: Using Pi Audio to Get Sound into Your Computer, Recording the Sound, Speaking to Your Pi, Making Movies, Adding Computer Vision Features with Open CV, Creating Live Streams.
8.	18B11CI313	Database Management Systems	CO1. Explain the characteristics, architecture of database approach, its components, different data models and the examples of their usage.

			CO2. For a given query write relational algebra expressions for that query and optimize the developed expressions.
			CO3. For a given specification of the requirement, design the databases using E-R method and normalization.
			CO4. Determine the functional dependency between two or more attributes, compute the closure of a set of attributes, evaluate a proposed decomposition
			CO5. Give examples of the application of primary, secondary, and clustering indexes, explain the theory and application of internal and external hashing techniques.
			CO6. Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.
			CO1. Familiarize with the security in databases and gaining familiarity with other popular databases used in the industry
9.	18B11CI311	Object Oriented System and Programming	CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors
			CO2: To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions.
			CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance.
			CO4: To learn the concept of Abstract classes and interfaces
			CO5: To learn the concepts of Operator overloading and conversion function
			CO6: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file.
			CO7: To learn the Exception Handling: try --catch and finally block, making user-defined exceptions.
			CO8: To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.
10.	18B11CI311	Object Oriented System and Programming	CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors

			CO2: To learn the designing of complex classes: Friend Function sand Static member functions, Inline functions, constant functions.
			CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance.
			CO4: To learn the concept of Abstract classes and interfaces
			CO5: To learn the concepts of Operator overloading and conversion function
			CO6: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file.
			CO7: To learn the Exception Handling: try-catch and finally block, making user-defined exceptions.
			CO8: To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.
11.	18B17CI372	IT Workshop (SciLab/MATLAB) Lab	CO1: Writing fundamental programs in Matlab, creating variables and mathematical functions
			CO2: Programming the fundamentals concepts of basic Plotting consisting of simple and multiple data sets in one plot
			CO3: Understand how to program matrix operations, array operations and how to solve the system of linear equations
			CO4: Understand how to program M-file scripts, M-file functions, Input–output Arguments
			CO5: Program control flow operators, loops, flow structures and debugging M-files
12.	18B11CI371	Object Oriented System and Programming Lab	CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors
			CO2: To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions.
			CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance.

SNo	Course Codes	Course Name	Course Outcomes
			CO4: To learn the concept of Abstract classes and interfaces. CO5: To learn the concepts of Operator overloading and conversion function. CO6: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file. CO7: To learn the Exception Handling: try—catch and finally block, making user-defined exceptions.
13.	18B17CI375	Python programming with Raspberry Pi Lab	CO-1 Learn about basics of Python programming . CO-2 Learn how to use Python and Shell scripts to control GPIO ports on the Pi CO-3 Understanding Python Programing and GPIO IN Pi3 CO-4 Understanding interfaces on a Pi 3 (I2C, SPI, etc.) CO-5 Storing sensor data in a dBase and plotting it CO-6 MQTT Protocol
14.	18B17CI373	Database Management Systems Lab	CO1: Design and implement a data base schema CO2: Designdifferentviewsoftablesfordifferentusersandtoapplyembeddedandnested queries CO3: Understand the use of structured query language and its syntax, transactions, data base recovery and techniques for query optimization CO4: Understand, analyze and apply common SQL statements including DDL, DML, DCL statements to perform different operations CO5: Develop application programs using PL/SQL CO6: Design and implement a project using embedded SQL and GUI

Semester: IV

SNo	Course Codes	Course Name	Course Outcomes
1.	10B11CI401	Microprocessors and Controllers	CO1: Knowledge of standard 32-bit Intel Architectures. CO2: Understanding of the Intel processors' seldom-used capabilities. CO3: Increase in proficiency with using C/C++ and assembler language. CO4: Learn to access service-functions provided in BIOS firmware.

			CO5: Gaining practical experience in programming peripheral I/O devices. CO6: Acquiring the background for understanding next-generation CPUs. CO7: To learn microprocessor programming model at a level that enables riting assembly language programs for the processor meeting given specifications. CO8: Learn concepts associated with interfacing a microprocessor to memory and to I/O devices. CO9: Learn how to control components of a microprocessor-based system through the use of interrupts.
2.	10B11CI411	Fundamentals of Algorithms	CO1: To Strengthen higher level cognitive skills of analysis, creation and evaluation. CO2: To Strengthen ability of data abstraction and problem-solving using computers. CO3: To Strengthen ability to express solutions to problems clearly and precisely. CO4: To Strengthen ability to design and evaluate ADTs, non-linear temporary and persistent data structures and also related algorithms. CO5: To Introduce students to some domain specific data structures and related algorithms in various domains.
3.	10B17CI407	Microprocessors and Controllers Lab	CO1: You will increase your proficiency with using assembler language, MASM. CO2: You will know how to access service-functions provided in BIOS firmware. CO3: You will gain practical experience in programming peripheral I/O devices. CO4: You will acquire the background for understanding next-generation CPUs. CO5: You will learn a microprocessor programming model at a level that enables you to write assembly language programs for the processor meeting given specifications. CO6: You will learn concepts associated with interfacing a microprocessor to memory and to I/O devices. CO7: You will learn how to control components of a microprocessor-based system through the use of interrupts.
4.	10B17CI471	Algorithms Lab	CO1: 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. CO2: Learn to program for sorting algorithms. Also, they will learn to implement Priority Queue and order statistics. CO3: Students will learn to implement searching techniques through RB-Tree and Skip list and hashing. CO4: Students will learn to implement basic graph algorithms (BFS, DFS, Prims, Kruskal, Dijkstra) and apply those to explore graph properties. CO5: Students will learn to implement algorithms for pattern matching. CO6: Students will learn to implement Dynamic

			programming methods for problems like, Longest common subsequence, Matrix chain multiplication, Coin denomination, Knapsack and back tracking algorithm to find a solution of the 3-SAT problem.
5.	19B11CI411	Software Engineering Practices	CO1: Describe the relative advantages and disadvantages among several major process models (e.g., waterfall, iterative, and agile). CO2: Describe the different practices that are key components of various process models. CO3: Differentiate among the phases of software development. CO4: Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system. CO5: Use a common, non-formal method to model and specify the requirements for a medium-size software system. CO6: Translate into natural language a software requirements specification (e.g., a software component contract) written in a formal specification language.
6.	18B11CI415	Object Oriented Programing	CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors. CO2: To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions. CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance. CO4: To learn the concept of Abstract classes and interfaces. CO5: To learn the concepts of Operator overloading and conversion function. CO6: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file. CO7: To learn the Exception Handling: try --catch and finally block, making user-defined exceptions. CO8: To learn the Unified Modeling Language (UML): Use Case Diagrams, State CO-8 Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.
7.	18B11CI411	Operating Systems	CO1: An appreciation of the role of an operating system. CO2: Create processes and threads. CO3: Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time. CO4: 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. CO5: Design and implement file management system. CO6: 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.
8.	18BI7CI471	Operating System Lab	CO1: Be able to create sockets and analyze different (client/server) models. CO2: Be able to create processes, threads, semaphores. CO3: Be able to analyze different protocols. CO4: Be able to learn how resources are being managed in Operating system. CO5: Be able to manage system memory
9.	19BI7CI471	Software Engineering Practices Lab	CO1: Students will be capable to acquire the generic software development skill through various stages of software life cycle. CO2: He will also be able to ensure the quality of software through software development with various protocol-based environment. After completion of course student will be able to generate test cases for software testing. CO3: Students will also be able to handle software development models through rational method. CO4: Rational Rose Enterprise Edition software is used to serve the objectives. CO5: The courses contains Basic Structural Modeling, Advance Structural Modeling, Basic Behavioral Modeling, Advance Behavioral Modeling, Architectural Modeling, UML Notation, UML Stranded Elements, Designing Test cases, Test Suits, Rational Unified Process etc.
10.	18B17CI474	Web Tech Lab	CO1: Basic PHP Concepts, PHP Operators, PHP Function, PHP Variables and Super globals. CO2: Conditional Statements, Looping Statements, Array, Cookies, PHP Form, PHP Session, File Upload, File Handling, User login and Registration. CO3: Database Connectivity, MySQL, MySQL connect, create DB/Table, Instructions such as select, where, order By, update and delete etc., encryption methods. 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. 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. CO6 Validate the registration, user login, user profile and payment by credit card pages using JavaScript.
11.	18B11CI414	Discrete Computational Mathematics	CO1: Students will be able to express logical statements in terms of logical connectives, predicates and quantifiers. CO2: Students will be able to apply various proving

				techniques such as direct, indirect proofs, mathematical induction, etc. CO3: They will learn basic set operations along with relations & functions with their types and usage. CO4: They will be familiar with graph & tree terminologies along with their various applications in computer science. CO5: Students will be able to solve counting problems using permutation, combinations techniques. CO6: They will learn about algebraic structures such as group, abelian group, rings, integral domain, fields, etc. CO7: Students will be able to analyze and solve various algorithms using recurrence relation methods.
12.	18B11CI474	Object Oriented Programing Lab		CO1: To learn the concepts of Objects, Classes, Methods, Constructors and Destructors CO2: To learn the designing of complex classes: Friend Functions and Static member functions, Inline functions, constant functions. CO3: To learn Inheritance: Single Inheritance, Multiple Inheritance, Multi-level Inheritance, Hierarchical Inheritance and Hybrid Inheritance. CO4: To learn the concept of Abstract classes and interfaces. CO5: To learn the concepts of Operator overloading and conversion function. CO6: To learn File Handling. Writing and reading data from the file, reading and writing the objects into the file. CO7: To learn the Exception Handling: try --catch and finally block, making user- defined exceptions. CO8: To learn the Unified Modeling Language (UML): Use Case Diagrams, State Diagrams, Sequence Diagrams, Communication Diagrams, and Activity Diagrams.
13.	18B17CI473	Data Simulation Lab		CO1: Student will learn simulation of real world problems using python, scipy and simpy. CO2: Students will learn structural development of complex system in terms of process, resources and levels. CO3: Student will learn to use random number generator. CO4: Students will learn to monitor and tally simulation results. CO5: Students will apply simulation and modelling techniques in many real examples. CO6: Students will apply simulation and modelling techniques in a real life project.
14.	18B11CI413	Modeling and Simulation Techniques		CO1: To learn the basic concepts, applications and terminology of computer simulation and modeling. CO2: To learn statistical methods of estimation and testing and other relevant concepts. CO3: To explain the working and applications of different types of simulation such as Monte Carlo, VS. Discrete Event. CO4: You will learn how to model a system and the

			execution of simulation tools. CO5: You will learn to analyze input data, its parameters, and the use of random number in a typical simulation study. CO6: Student will learn different techniques for the Verification and Validation of a simulation study.

Semester: V

SNo	Course Codes	Course Name	Course Outcomes
1.	10B11CI511	Operating Systems	CO1: An appreciation of the role of an operating system. CO2: Understand the theory and logic behind the design and construction of operating systems. CO3: Examine the algorithms used for various operations on operating systems. CO4: Differentiate between various operating systems functionalities in terms of performance. CO5: Know the problems in the design of operating system and study the probable solutions. CO6: Become aware of the issues in the management of resources like processor, memory and input-output.
2.	10B17CI571	Operating Systems Lab	CO1: Be able to create sockets and analyze different (client/server) models. CO2: Be able to create processes, threads, semaphores. CO3: Be able to analyze different protocols. CO4: Be able to learn how resources are being managed in Operating system. CO5: Be able to manage system memory
3.	10B17CI572	Software Engineering Lab	CO1: Students will be capable to acquire the generic software development skill through various stages of software life cycle. CO2: He will also be able to ensure the quality of software through software development with various protocol based environment. After completion of course student will be able to generate test cases for software testing. CO3: Students will also be able to handle software development models through rational method. CO4: Rational Rose Enterprise Edition software is used to serve the objectives. CO5: The courses contains Basic Structural Modeling, Advance Structural Modeling, Basic Behavioral Modeling, Advance Behavioral Modeling, Architectural Modeling, UML Notation, UML Stranded Elements, Designing Test cases, Test Suits, Rational Unified Process etc.
4.	10B11CI512	Software Engineering	CO1: Describe the relative advantages and disadvantages among several major process models (e.g., waterfall, iterative, and agile). [Familiarity]

			CO2: Describe the different practices that are key components of various process models. [Familiarity] CO3: Differentiate among the phases of software development. CO4: Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a medium-sized software system. [Usage] CO5: Use a common, non-formal method to model and specify the requirements for a medium-size software system. [Usage] CO6: Translate into natural language a software requirements specification (e.g., a software component contract) written in a formal specification language. [Usage]
5.	10B1WCI515	Software Testing and Debugging	CO1: You will broaden your knowledge of software engineering. CO2: You will learn Software testing algorithms and programs. CO3: You will increase your proficiency in JAVA Language. CO4: You will know how strategies and tactics of effective and efficient testing. CO5: You will gain practical experience in design, develop, and document static, white-box, black-box tests. CO6: You will acquire the background for understanding Test Management and Software Development. CO7: You will acquire the knowledge of higher order and object oriented testing. CO8: You will learn a effective and efficient use of debugging techniques. CO9: You will learn how to make use of Web testing and Automated software testing.
6.	10B1WCI575	Software Testing and Debugging Lab	CO1: Have an ability to apply software testing knowledge and engineering methods. CO2: Have an ability to design and conduct a software test process for a software testing project. CO3: Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation. CO4: 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. CO5: Have an ability to use various communication methods and skills to communicate with their teammates to conduct their practice-oriented software testing projects. CO6: Have basic understanding and knowledge of contemporary issues in software testing, such as component-based software testing problems.

			CO7: Have an ability to use software testing methods and modern software testing tools for their testing projects.
7.	10B22CI521	Web Application Engineering	CO1: The student shall acquire the skill to design and develop webbased applications with high usability, scalability and efficiency. CO2: They shall be exposed to various technologies required to design web sites CO3: They shall acquire the skill to choose the technology to use based on the requirements and functionality of the web site.
8.	10B28CI581	Web Technology Lab	CO1: Basic PHP Concepts, PHP Operators, PHP Function, PHP Variables and Super globals. CO2: Conditional Statements, Looping Statements, Array, CO3: Cookies, PHP Form, PHP Session, File Upload, File Handling, User login and Registration. CO4: Database Connectivity, MySQL, MySQL connect, create DB/Table, Instructions such as select, where, order By, update and delete etc., encryption methods. CO5: 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. CO6: 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. CO7: Validate the registration, user login, user profile and payment by credit card pages using JavaScript.
9.	18B1WCI535	Python Programming	CO1Familiarity about concepts of Python Programming. Broaden the knowledge about Variables, expressions and Functions in Python. CO2Broaden the knowledge about Branching and Iteration. To have hands on skillson String Manipulation, Guess and Check, Approximations, Bisection CO3To learn about Decomposition Abstractions, Tuples, Lists, Dictionaries and Illustrative programs: CO4 Files, Modules, Packages and Testing, Debugging, Exceptions, Assertions: CO5Understanding and analyzing Object Oriented Programming: CO6To have hands on skills on Illustrative programs (examples Sorting and Searching, Regular expressions) and GUI

10	18B1WCI533	Principle of Programming Language	CO1: Tolearnmajorprogrammingparadigmsandtechniques involvedindesignandimplementationofmodernprogram minglanguages.Tolearnthestructureofacompiler and interpretation. To learn syntax and semantic of programming language.
			CO2: To learn the structured and object-oriented programming paradigm.
			CO3: 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).
			CO4: To learn the concurrency in programming languages, Exception handling and Scripting languages
11	18B1WCI573	Principle of Programming Language Lab	CO1: Tolearnmajorprogrammingparadigmsandtechniques involvedindesignandimplementationofmodernprogram minglanguages.To learn the structure of a compiler and interpretation. To learn syntax and semantic of programming language.
			CO2: To learn the structured and object-oriented programming paradigm.
			CO3: 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).
			CO4: To learn the concurrency in programming languages, Exception handling and Scripting languages
12	18B1WCI534	Java Programming	CO1: 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, check boxes, text fields, scrollbars and scrolling lists, using Abstract Windowing Toolkit (AWT)&Swings
			CO2: To learn Java Data Base Connectivity (JDBC)so as to retrieve and manipulate the information on any relational database through Java programs.
			CO3: To learn the server-side programming using Servlets and JSP.
			CO4: To learn the invocation of the remote methods in an application using RMI
13	18B1WCI574	Java Programming Lab	CO1: 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
			CO2: To learn Java Data Base Connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through Java programs. CO3: To learn the server side programming using Servlets and JSP. CO4: To learn the invocation of the remote methods in an application using RMI
14	18B11CI514	Computer Organization & Architecture	CO-1 To learn the basic concepts, terminology and evolution in computer organization and architecture. CO-2 Understanding the computer architecture and computer arithmetic. CO-3 Understanding of the computer memory and the issues related to memory. CO-4 Understanding the concept of memory I/O, interrupt handling and DMA. CO-5 Learn the organization of Processor and the concept of pipelining. CO-6 Learning concepts of Parallel processing and related issues.
15	18B17CI574	Computer Organization & Architecture Lab	CO-1 Ability to understand basic structure of computer. CO-2 Ability to perform arithmetic operations on computer. CO-3 Ability to understand the memory concepts including Cache. CO-4 Familiarity with CPU design.
16	18B11CI511	Advanced Java	CO-1 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 CO-2 To learn Java Data Base Connectivity (JDBC) so as to retrieve and manipulate the information on any relational database through Java programs. CO-3 To learn the server side programming using Servlets and JSP CO-4 To learn client-server networking programming. CO-5 To learn the invocation of the remote methods in an application using RMI CO-6 To learn the development of Enterprise based applications, using EJB: Stateful, Stateless and Entity Beans, Using Struts and Hibernate Frameworks.
17	18B17CI571	Advanced Java Lab	CO-1: Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing and Event Handling. CO-2: Design and develop dynamic web applications using Servlets and JSP. CO-3 Designing client-server networking applications. CO-4 Designing distributed applications using RMI. CO-5: Designing Enterprise based applications by encapsulating an application's business logic. CO-6: Designing applications using pre-built frameworks.
18	18B11CI512	Information	CO 1: To understand the fundamental terms and concepts of

		Systems	information systems, associated technologies and underlying infrastructure. CO-2: To understand how IT provides value by improving the management process through access to better information. CO-3: To acquire knowledge and develop hands on skills on various information system development technologies. CO-4: To understand and analyze information system categories along with their principles and design issues. CO-5: To understand and analyze the social issues and ethical implications of technology across organizations and society. CO-6 To develop the ability to evaluate information systems, assess user needs, propose solutions, and evaluate proposals for implementation.
19	18B17CI572	Information Systems Lab	CO1: To analyze how IT provides value by improving the management process through access to better information. CO2: To understand and apply system development and project management principles. CO3: To acquire knowledge of various information system development technologies. CO4: To have hands on skills to evaluate Servlet technology for information development. CO5: To have hands on skills to evaluate JSP technology for information development. CO6: To develop the ability to evaluate information systems, assess user needs, propose solutions, and evaluate proposals for implementation.
20	18B1WCI532	Data Compression	CO-1: To understand the statistical basis for and performance metrics for lossless compression. CO-2: To understand the conceptual basis for commonly used lossless compression Techniques. CO-3: To understand how to use and evaluate several readily available implementations of those techniques. CO-4: To understand the structural basis for and performance metrics for commonly used lossy compression techniques. CO-5: To understand the conceptual basis for commonly used lossy compression techniques. CO-6: To implement graph theory in compression methodologies for images in MATLAB. CO-7: To understand image compression techniques case studies.
21	18B1WCI572	Data Compression Lab	CO-1 To understand the statistical basis for and performance metrics for lossless compression. CO-2 To understand the conceptual basis for commonly used lossless compression techniques CO-3 To understand how to use and evaluate several readily available implementations of those techniques. CO-4 To understand the structural basis for and

ocr stamp

				performance metrics for commonly used lossy compression techniques. CO-5 To understand the conceptual basis for commonly used lossy compression techniques CO-6 To implement graph theory in compression methodologies for images in MATLAB. CO-7 To understand image compression techniques" case studies.
22	20B1WCI531	Foundation for Data Science and visualization		CO-1: Understanding the basics of data science CO-2 Using versatile and flexible languages (Python and R programming) for supporting data science. CO-3 Using data processing for collecting and manipulating the data into the usable and desired form Usage CO-4 Using data visualization to easily access the huge amount of data in visuals Usage CO-5 Using statistics to collect and analyze the numerical data in a large amount and finding meaningful insights from i Usage CO-6 Understanding linear algebra to represent, model, synthesize and summarize the complex data
23	20B1WCI571	Data Science and Visualization Lab		CO-1 Understanding the basics of data science. CO-2 Using versatile and flexible languages (Python and R programming) for supporting data science. CO-3 Using data processing for collecting and manipulating the data into a usable and desired form. CO-4 Using statistics to collect and analyze the numerical data in a large amount and finding meaningful insights from i. CO-5 Understanding linear algebra to represent, model, synthesize and summarize the complex data. CO-6 Using data visualization to easily access the huge amount of data in visuals.
24	20B1WCI532	Cloud Computing: Concepts, Technology & Architecture		CO-1 To learn the basic concepts, applications, and terminology of cloud computing. CO-2 To learn basic concepts of infrastructure management and load balancing. CO-3 To learn basic concepts of cloud security and metrics for evaluation. CO-4 To understand cloud computing and its role in new distributed computing implementation.
25	20B1WCI572	Cloud Computing: Concepts, Technology & Architecture Lab		CO1 To learn the basic concepts, applications, and terminology of cloud computing. CO2 To learn basic concepts of infrastructure management and load balancing. CO3 To learn basic concepts of cloud security and metrics for evaluation. CO4 To understand cloud computing and its role in new

| | | | distributed computing implementation. |

Semester: VI

SNo	Course Codes	Course Name	Course Outcomes
1.	10B11CI611	Computer Networks	CO1: To learn the basic concepts and terminology in computer networks. CO2: To learn about the layered models in computer networks and different types of network topologies and protocols. CO3: To learn about the data link layer and MAC layer protocols and related issues. CO4: 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. CO5: To learn about the transport layer protocols and related issues. CO6: You will learn about the session, presentation and application layers protocols. CO7: Further, to learn about the some advanced topics in networks such as Cryptographic algorithms, Network security and management, and concepts of wireless networks.
2.	10B11CI612	Compiler Design	CO1: To understand the basic concept of compilation particular, lexical analyzer, syntax and semantic analysis, code generation and optimization phases of compilation. CO2: Ability to create lexical rules and grammars for a programming language. CO3: Ability to use flex or similar tools to create a lexical analyzer and Yacc/Biscon tools to create a parser. CO4: Ability to implement a various parser such as a bottom-up SLR parser without using any compiler-generation tools. CO5: Ability to implement a various intermediate code generation techniques without using any compiler generation tools. CO6: Ability to implement various code optimizing techniques without using any compiler-generation tools CO7: Ability to implement semantic rules into a parser that performs attribution while parsing CO8: Ability to design a compiler for a concise programming language.
3.	10B11CI613	Computer Organisation and Architecture	CO1: To learn the basic concepts, terminology and evolution in computer organization and architecture. CO2: Understanding the computer architecture and computer arithmetic. CO3: Understanding of the computer memory and the

				issues related to memory. CO4: Understanding the concept of memory I/O, interrupt handling and DMA. CO5: Learn the organization of Processor and the concept of pipelining. CO6: Learning concepts of Parallel processing and related issues.
4.	10B11CI673	Computer Organization & Architecture Lab		CO-1 Ability to understand basic structure of computer. CO-2 Ability to perform arithmetic operations on computer. CO-3 Ability to understand the memory concepts including Cache. CO-4 Familiarity with CPU design.
5.	11B1WCI611	Computer Graphics		CO1: Student will learn about the overview of computer graphic applications and graphics devices (Display Technologies, Raster Refresh (Raster-Scan), CRT, LCD displays, etc.) CO2: Student will learn about the scan conversion - lines, circles and Ellipses, filling, clipping and aliasing. CO3: 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. CO4: Student will learn about the Three-Dimensional transformations and viewing in 3D. CO5: 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.
6.	10B17CI671	Computer Networks Lab		CO1: To understand the working concepts of Networking and inter – networking Devices. CO2: To understand the concepts of different shortest path algorithms. CO3: To understand different error detection and correction techniques/algorithms. CO4: To understand Flow control techniques/algorithms. CO5: To understand the concepts of client – server interaction using connection oriented and connectionless protocols. CO6: To understand the proficiency in Traffic Shaping Algorithms. CO7: You shall be exposed to working of encryption and decryption algorithms.
7.	10B17CI672	Compiler Design Lab		CO1: Gain an in-depth understanding of the principles underlying the design. CO2: Construction of compilers. CO3: Functioning of Compiler writing tools. CO4: Building various parsing techniques.

8.	10B17CI673	System and Network Programming Lab	CO1: Student will be able to create sockets and analyze different (client/server) models. CO2: Student will be able to create processes, threads, semaphores and Bluetooth programming. CO3 Student will be able to analyze different protocols.
9.	11B1WCI671	Computer Graphics Lab	CO1: Using OpenGL for Graphics. CO2: Programming User-Interface Issues. CO3: Concepts of 2D & 3D object representation. CO4: Implementation of various scan and clipping algorithms. CO5: 2 D Modelling.
10	18B1WCI634	Machine Learning	CO1: To learn the basic concepts and terminology in machine learning. CO2: To learn about the definition of learning systems, their goals and applications in machine learning. CO3: To understand concepts associated with classification and experimental evaluation of classification algorithms. CO4: To learn concepts associated with decision trees and experimental evaluation of classification algorithms. CO5: To learn about instance-based learning, clustering and unsupervised learning.
11	19B1WCI631	Digital Forensics	CO1: Introduction to Cyber Crime and Ethical Hacking. CO2: Introduction to Digital Forensics and Digital Evidences. CO3: Computer Security Incident Response Methodology. CO4: Forensic Duplication and Disk Analysis, and Investigation Data Analysis. CO5: Network Forensics Incidents, Using Routers as Response Tools. CO6: Forensic Investigation Report and Forensic Tools.
12	18B1WCI635	Data Mining & Data Warehousing	CO1: To describe the concept of Data warehouse & its attributes. CO2: To study different data warehouse models, architectures and implementation. CO3: To understand the basic concept of data mining and its functionality. CO4: To understand the concept of classification techniques and its implementation.

				CO5: To understand the concept of association rules, different techniques and implementation details. CO6: To understand the concept of cluster analysis, anomaly detection and its usage and implementation details.
13	19B1WCI632		Information Security	CO1: Understand the need for security. CO2: Analysis of security algorithms. CO3: Demonstrate familiarity with major security algorithms and data structure. CO4: Apply important security algorithmic design paradigms and methods of analysis.
14	19BIWCI637		Statistics And Exploratory Data Analytics	CO1: Analyze the concept of statistics and exploratory data analysis. CO2: Graphical and modeling techniques for exploring data. CO3: Implement data transformation, dimensionality reduction. CO4: Apply data clustering and visualized exploratory data through various plots and graphs.
15	19BIWCI635		Architecting Distributed Cloud Applications	CO1: Introduction to cloud computing and distributed cloud applications. CO2: Network Communication in distributed cloud applications. CO3: Message Communication in distributed cloud applications. CO4: Versioning, upgrading, and configuration of distributed cloud applications.
16	18B11CI611		Computer Networks	CO1: To learn the basic concepts and terminology in computer networks. CO2: To learn about the layered models in computer networks and different types of network topologies and protocols. CO3: To learn about the data link layer and MAC layer protocols and related issues. CO4: 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. CO5: To learn about the transport layer protocols and related issues. CO6: You will learn about the session, presentation and application layers protocols.

				CO7: Further, to learn about the some advanced topics in networks such as Cryptographic algorithms, Network security and management, and concepts of wireless networks.
17	18B17CI671	Computer Networks Lab		CO1: To understand the working concepts of Networking and inter – networking Devices. CO2: To understand the concepts of different shortest path algorithms. CO3: To understand different error detection and correction techniques/algorithms. CO4: To understand Flow control techniques/ algorithms. CO5: To understand the concepts of client – server interaction using connection oriented and connectionless protocols. CO6: To understand the proficiency in Traffic Shaping Algorithms. CO7: You shall be exposed to working of encryption and decryption algorithms.
18	10B22CI622	Data Mining		CO1: To describe the concept of Data warehouse & its attributes. CO2: To study different data warehouse models, architectures and implementation. CO3: To understand the basic concept of data mining and its functionality. CO4: To understand the concept of classification techniques and its implementation. CO5: To understand the concept of association rules, different techniques and implementation details. CO6: To understand the concept of cluster analysis, anomaly detection and its usage and implementation details.
19	10B28CI682	Data Mining Lab		CO1: Practical exposure on implementation of well-known data mining tasks. CO2: Exposure to real life data sets for analysis and prediction. CO3: Learning performance evaluation of data mining algorithms in a supervised and an unsupervised setting. CO4: Handling a small data mining project for a given practical domain. CO5: Develop and apply machine learning tools and techniques.
20	18B1WCI674	Machine Learning Lab		CO1: To implement classification algorithms in python. CO2: To implement Clustering algorithms in python. CO3: To implement Genetic Algorithms in Python. CO4: Top compare different algorithms based on some common factors.
21	19B1WCI671	Digital Forensics Lab		CO1: Introduction to Cyber Crime and Ethical Hacking.

			CO2: Introduction to Digital Forensics and Digital Evidences.
			CO3: Computer Security Incident Response Methodology.
			CO4: Forensic Duplication and Disk Analysis, and Investigation Data Analysis.
			CO5: Network Forensics Incidents, Using Routers as Response Tools.
			CO6: Forensic Investigation Report and Forensic Tools.
22	18B1WCI675	Data Mining & Data Warehousing Lab	CO1: Practical exposure on implementation of well-known data mining tasks.
			CO2: Exposure to real life data sets for analysis and prediction.
			CO3: Learning performance evaluation of data mining algorithms in a supervised and an unsupervised setting.
			CO4: Handling a small data mining project for a given practical domain.
			CO5: Develop and apply machine learning tools and techniques.
23	19B1WCI672	Information Security Lab	CO1: Understand the basic security services e.g. Authentication, Access Control, Confidentiality, Integrity.
			CO2: Learn standard symmetric encryption algorithms.
			CO3: Learn architecture for public and private key cryptography.
			CO4: Learn the methods of digital signature and encryption.
			CO5: Learn key management and how key exchange protocols work.
			CO6: Learn futuristic cryptographic techniques like Elliptic Curve and quantum cryptography.
24	19B1WCI677	Statistics And Exploratory Data Analytics Lab	CO1: Understand the basics of MATLAB syntax/python libraries.
			CO2: Exploring dataset and visualizing data on different charts.
			CO3: Perform the basic operations of data pre-processing and dimensionality reduction.
			CO4: Design and analyze clustering techniques.
			CO5: Understanding of curve fitting toolbox.

SNo	Course Codes	Course Name	Course Outcomes
			CO6: Implementation of dendrogram and treemaps
25	19B1WCI675	Architecting Distributed Cloud Applications Lab	CO1: Introduction to cloud computing and distributed cloud applications. CO2: Network Communication in distributed cloud applications. CO3: Message Communication in distributed cloud applications. CO4: Versioning, upgrading, and configuration of distributed cloud applications.
26	18B19CI691	Minor Project	CO1: Identify an open ended problem in area of mechanical engineering which requires further investigation. CO2: Identify the methods and materials required for the project work. CO3: Manage the work with team members. CO4: Formulate and implement innovative ideas for social and environmental benefits. CO5: Analyze the results to come out with concrete solutions. 6. Write technical report of the project apart from developing a presentation.

Semester: VII

SNo	Course Codes	Course Name	Course Outcomes
1	20B1WCI732	From Graph to Knowledge Graph	CO1: Learn graph representations. CO2: Utilize fundamentals to build knowledge graphs. CO3: Understand the algorithms and techniques for crawling web sites, structured data extraction, and information extraction from unstructured text. CO4: Understand how to apply the tools and infrastructure to build and query knowledge graphs.
2	20B1WCI772	From Graph to Knowledge Graph Lab	CO1: Build a knowledge graph from documents. CO2: Building a knowledge graph from a text. CO3: Build a Small Knowledge Graph. CO4: Understand the Graph Databases. CO5: Familiarity with graph computing Frameworks. CO6: Able to Visualization of graphs
3	13B1WCI731	Arm Based Embedded System Design	CO1: Describe the architecture of a typical embedded RISC processor (e.g. ARM Cortex-M3). CO2: Develop an understanding of the instruction set and addressing modes. CO3: Exercise a range of typical microcontroller peripherals (e.g. GPIO, USART, ADC, etc...). CO4: Use a typical toolchain to implement and test simple

				embedded microcontroller applications in C and assembly language. CO5: Evaluate the requirements for embedded Real Time Operating Systems (RTOS). CO6: Understand the structure of a RTOS (e.g. eLinux)
4	19B1WCI734	Object Oriented Technologies Using Java		CO1: Understanding the features and concepts of Object-Oriented Programming using Java. CO2: Defining classes, objects, constructors, methods in Java. CO3: Inheritance, Interfaces, and Polymorphism. CO4: Packages, Exception Handling, and Multithreading.
5	19B1WCI774	Object Oriented Technologies Using Java Lab		CO1: Understanding the features and concepts of Object-Oriented Programming using Java. CO2: Defining classes, objects, constructors, methods in Java. CO3: Inheritance, Interfaces, and Polymorphism. CO4: Packages, Exception Handling, and multithreading.
6	18B1WCI734	Cryptography& network security		CO-1 Understand the basic security services e.g.Authentication, Access Control, Confidentiality, Integrity, and Non repudiation). CO-2 Learn standard symmetric encryption algorithms CO-3 Learn the architecture for public and private key cryptography and how public key infrastructure (PKI) supports network security. CO-4 Learn the methods of digital signature and encryption CO-5 Learn key management and how key exchange protocols work. CO-6 Learn futuristic cryptographic techniques like Eliptic Curve and quantum cryptography.
7	18B1WCI742	Artificial Intelligence		CO-1 Determine the characteristics of a given problem that an intelligent system must solve. CO-2 Apply Bayes rule to determine the probability of a hypothesis given evidence. CO-3 Identify examples of knowledge representations for reasoning under uncertainty. CO-4 List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised. CO-5 Identify examples of classification tasks, including the available input features and output to be predicted.
8	18B1WCI732	Social and Information Network Analysis		CO-1. Understand what constitutes a social network CO-2. Represent networks in graph-theoretic language CO-3. Design effective and reliable network research projects CO-4. Configure network and attribute data for standard software packages CO-5. Comprehend the reasons for using, and principles of, permutation tests CO-6. Apply centrality measures appropriatelyy

9	18B1WCI772	Artificial Intelligence Lab	CO-1 Determine the characteristics of a given problem that an intelligent system must solve. CO-2 Apply Bayes' rule to determine the probability of a hypothesis given evidence. CO-3 Identify examples of knowledge representations for reasoning under uncertainty. CO-4 List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised. CO-5 Identify examples of classification tasks, including the available input features and output to be predicted.
10	20B17CI771	Advanced Software Systems Lab	CO-1 Knowledge and Understanding of the lists, exception handling, dictionaries, and NumPy arrays in python. CO-2 Knowledge and Understanding of scatter plots and inferring the information from scatter plots. CO-3 Understanding and implementing Probability functions. CO-4 Understanding and generating Area Under a Curve (AUC) for the given discrete or continuous functions and generating PDF curves. CO-5 Understanding to import CSV datasets. Applying statistical functions on the dataset and classifying dataset. Building machine learning models. CO-6 Understanding and implementing data wrangling, Advanced graphs and 3-D graphs.
11	10M11CI112	Advanced Computer Networks	CO1: Understand the main abstract concepts related to the layered communication architecture Cognitive skills (thinking and analysis). CO2: Analyze and implement some of the most advanced routing and congestion control algorithms. CO3: Evaluate the performances of computer networks (through mathematical modeling and simulation) Communication skills (personal and academic). CO4: Understand basics and principles of new generation of computer networks (VPN, wireless networks, mobile networks...). CO5: Practical and subject specific skills (Transferable Skills).
12	10M11CI114	High Performance Computer Architecture	CO1: To evaluate and analyze cost and performance of multi-processor systems. CO2: To develop and simulate assembly language programs on pipelined and serial architecture. CO3: To develop, implement, and demonstrate the learning through a project that meet stated specifications. CO4: To understand and be able to explain different parallel architectures, interconnections and various memory organization in modern high-performance architectures.

13	12B1WCI734	C# and VB.NET	CO1: Learn about MS.NET framework developed by Microsoft. CO2: You will be able to using XML in C#.NET specifically ADO.NET and SQL server CO3: Be able to understand use of C# basics, Objects and Types, Inheritance CO4: To develop, implement and creating Applications with C#. CO5: To develop, implement, and demonstrate Component Services, Threading, Remoting, Windows services, web CO6: To understand and be able to explain Security in the .NET framework and Deployment in the .NET. CO7: To develop Assemblies and Deployment in .NET, Mobile Application Development.
14	15B1WCI731	Mobile Computing	CO1: To acquire solid knowledge on mobile networks and mobile computing. CO2: To develop applications that are mobile-device specific and demonstrate current practice in mobile computing contexts. CO3: To understand the concepts of Adhoc and wireless sensor networks, and Mobile IP. CO4: To be able to analyze the performance of different handoff, roaming, and location update algorithms for cellular networks. CO5: To develop applications that are mobile-device specific and demonstrate current practice in mobile computing contexts.
15	10B1WCI733	Graph Algorithms and Applications	CO1: To present a rigorous introduction to the fundamentals of Graph Theory and Graph algorithms. CO:2 To enable the students to model various applications from Computer Science and Engineering using Graphs, CO:3 To introduce the techniques to store, manipulate and answer queries about a graph using a computer.
16	10M11CI113	Advanced Database Systems	CO1:To understand the basic Transaction Management in Databases. CO2: To understand the concepts of Query processing and Query Optimization. CO3: To know the important aspects of Single and Multilevel dynamic Indexing. CO4: To understand the Data warehouse and Data mining. CO5: To learn about how to find out the best query evaluation Plan. CO6: To learn Query Optimization Techniques. CO7: To understand how to apply appropriate known concurrency control techniques, recovery mechanism and query evaluation plan for a given scenario.
17	12B1WCI733	Advance JAVA	CO1: To learn the basic concepts of Internet Programming, using Java Applets

			CO2: To create a full set of UI widgets using Abstract Windowing Toolkit (AWT) & Swings CO3: Able to perform event handling on AWT and Swing components. CO4: Learn to access database through Java programs, using Java Data Base Connectivity (JDBC) CO5: To develop reusable software component, using Java Bean. CO6: Understanding the multi-tier architecture of web-based enterprise applications using Enterprise JavaBeans (EJB).
18	10B1WCI735	Network Security and Cryptography Techniques (OE)	CO1: Understand the basic security services e.g.Authentication, Access Control, Confidentiality, Integrity, and Non repudiation). CO2: Learn standard symmetric encryption algorithms CO3: Learn the architecture for public and private key cryptography and how public key infrastructure (PKI) supports network security. CO4: Learn the methods of digital signature and encryption CO5: Learn key management and how key exchange protocols work. CO6: Learn futuristic cryptographic techniques like Eliptic Curve and quantum cryptography.
19	10M11CI111	Advanced Data Structures	CO1: Analyze the asymptotic performance of algorithms. CO2: Write rigorous correction proofs of algorithms. CO3: Demonstrate a familiarity with major algorithms and data structure. CO4: Apply important algorithmic design paradigms and method of analysis.
20	13B1WCI731	ARM based Embedded System Design	CO1: To learn the basic concepts of Computer Architecture. CO2: To learn ARM Software Development Tools and ARM Architecture Fundamentals. CO3: Understanding ARM Assembly Language Programming Basics. CO4: To learn Programming Techniques for ARM.
21	11B1WC1832	Information Retrieval and Data Mining	CO1: To describe the concept of Data warehouse & its attributes CO2:To study different data warehouse models, architectures and implementation CO3:To understand the basic concept of data mining and its functionality CO4:To understand the concept of classification techniques and its implementation CO5:To understand the concept of association rules, different techniques and implementation details CO6:To understand the concept of cluster analysis, anomaly detection and its usage and implementation details

22	10B1WCI731	Artificial Intelligence	CO1:Demonstrate fundamental understanding of artificial intelligence (AI) and expert systems. CO2:Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning. CO3: Demonstrate proficiency in applying scientific method to models of machine learning. CO4: Discuss the awareness of ANN and different optimizations techniques CO5:Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning. CO6:Demonstrate proficiency in applying scientific method to models of machine learning. CO7:Discuss the basics of ANN and different optimizations techniques.
23	10B1WCI737	Image Processing	CO1: To understand how digital images are represented, manipulated, encoded and processed, with emphasis on algorithm design, implementation and performance evaluation. CO2: Emphasis will be to develop engineering skills and intuitive understanding of the tools used in Image Processing. CO3: The students would be encouraged to design and develop the image processing algorithms/tools to real life problems.
24	18B1WCI731	Python Programming	CO1: Familiarity about concepts of Python Programming. Broaden the knowledge about Variables, expressions and Functions in Python. CO2: Broaden the knowledge about Branching and Iteration. To have hands on skills on String Manipulation, Guess and Check, Approximations, Bisection CO3: To learn about Decomposition Abstractions, Tuples, Lists, CO4: Dictionaries and Illustrative programs: Files, Modules, Packages and Testing, Debugging, Exceptions, Assertions: CO5: Understanding and analyzing Object Oriented Programming: CO6: To have hands on skills on Illustrative programs(examples Sorting and Searching, Regular expressions) and GUI
25	10B19CI791	Project Part I	CO1: To Plan/design and code, dependable software/hardware. CO2: To act as catalyst in transferring the Computer Science and Engineering and Information Technology knowledge to field usage for the socio-economic development of the society.
26	10B1WCI731	Artificial	CO1: Demonstrate fundamental understanding of artificial

		Intelligence	intelligence (AI) and expert systems. CO2: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning. CO3: Demonstrate proficiency in applying scientific method to models of machine learning. CO4: Discuss the awareness of ANN and different optimizations techniques CO5: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning. CO6: Demonstrate proficiency in applying scientific method to models of machine learning. CO7: Discuss the basics of ANN and different optimizations techniques.
27.	18B1WCI731	Python Programming	CO1: Familiarity about concepts of Python Programming. Broaden the knowledge about Variables, expressions and Functions in Python. CO2: Broaden the knowledge about Branching and Iteration. To have hands on skills on String Manipulation, Guess and Check, Approximations, Bisection CO3: To learn about Decomposition, Abstractions, Tuples, Lists, Dictionaries and Illustrative programs: Files, Modules, Packages and Testing, Debugging, Exceptions, Assertions: CO4: Understanding and analyzing Object Oriented Programming: CO5: To have hands on skills on Illustrative programs(examples Sorting and Searching, Regular expressions) and GUI
28.	11B1WCI731	Software Agents	CO1: Student shall deepen the generic skills to design and implement software agents, software agent's tools, architectures for a broad-based set of user-computer and automating systems in various domains. CO2: Student will explore, study, evaluate, present several agent tools. CO3: Teaching in this course is designed to engage the students in active and experiential learning by taking a problem solving and design oriented approach with special emphasis on real world applications.
29.	18B1WCI733	Computational Techniques and Algorithms	CO1: Solve systems of linear equations using multiple methods, including Gaussian elimination and matrix inversion. Algorithms for vector and matrix operations. PositiveDefinite Systems, Cholesky Decomposition Decomposition, Sensitivity and round-off errors Least Squares Problem, OR Decomposition SVD and QR algorithm CO2: Determine Eigen values and eigenvectors and solve Eigen value problems Iterative algorithms and

			Convergence
30.	10B19CI792	Project Part- 1	CO-1 To Plan/design and code, dependable software/hardware. CO-2 To act as catalyst in transferring the Computer Science and Engineering and Information Technology knowledge to field usage for the socio-economic development of the society.
31.	18B1WCI733	Advanced Algorithms	CO-1 Analyze the asymptotic performance of algorithms. CO-2 Write rigorous correction proof s of algorithms. CO-3 Demonstrate a familiarity with major algorithms and data structure. CO-4 Apply important algorithmic design paradigms and method of analysis.
32.	18B1WCI773	Advanced Algorithms Lab	CO-1 Analyze the asymptotic performance of algorithms. CO-2 Write rigorous correction proof s of algorithms. CO-3 Demonstrate a familiarity with major algorithms and data structure. CO-4 Apply important algorithmic design paradigms and method of analysis.
33.	19B1WCI731	Computational Data Analysis	CO-1 To learn the basic concepts and terminology in statistics used in machine learning. CO-2 To understand the concepts associated with the classification and experimental evaluation of classification algorithms. CO-3 To learn about clustering and unsupervised learning. CO-4 To learn about various techniques for selecting relevant features selection. CO-5 To learn about various techniques for enhancing the performance of basic machine learning algorithms.
34.	19B1WCI771	Computational Data Analysis lab	CO1 To learn and implement the basic concepts and terminology in statistics used in machine learning. CO2 To implement classification and clustering algorithms in python. CO3 To implement feature selection methods in Python. CO4 To compare different algorithms based on some common factors.
35.	18B1WCI774	Cryptography& network security Lab	CO-1 Implement the cipher techniques. CO-2 Apply the mathematical foundation required for various cryptographic algorithms. CO-3 Develop the various security algorithms CO-4 Design the signature scheme by applying Digital Signature Standard. CO-5 Use different open source tools for network security and analysis. CO-6 Demonstrate the intrusion detection system
36.	18B1WCI736	Storage Networks	CO-1 Basics of Storage networks. CO-2 Design and Implement the RAID Levels. CO-3 Discussing algorithms related to storage networks. CO-4 Designing the SAN, NAS, CAS, and DAS based storage networks.

			CO-5 Concepts of Data Deduplication, and File Systems. CO-6Discussion of advanced topics of Distributed storage networks, protocols, and architecture.
37.	19B1WCI736	Information Auditing & Risk Management	CO-1To learn to explain why it is important to design interactive products that are usable, define key terms used in interaction design. CO-2To learn to explain key theories used in the design of interactive products and to explain the importance of iteration, evaluation, and prototyping in interaction design. CO-3To gather data in the context of developing a simple interactive product using suitable techniques. CO-4To produce a low-fidelity prototype for an interactive product based upon a simple list of interaction design principles and to evaluate an interactive product using suitable techniques. CO-5To communicate effectively to peers and specialists about requirements, design, and evaluation activities relating to interactive products. CO-6To define a suitable program of user involvement that treats users ethically and fairly.
38.	19B1WCI737	Optimization Methods in Business Analytics	CO-1 Comprehension of the design of the optimized algorithm. CO-2 Demonstrate familiarity with major algorithms. CO-3 Analyze the different types of algorithms. CO-4 Apply algorithms for simulating the decision.
39.	18B1WCI840	Computer Vision	CO1 Implement fundamental image processing techniques required for computer vision. CO-2 Perform shape analysis and implement boundary tracking techniques. CO-3 Apply chain codes and other region descriptors. CO-4 Apply Hough Transform for line, circle, and ellipse detections. CO-5Apply 3D vision techniques, implement motion-related techniques, and Develop applications using computer vision techniques.
40.	19B1WCI738	Introduction to Deep Learning	CO-1 Linear regression: mean squared error, analytical solution. CO-2 Logistic regression: model, cross-entropy loss, class probability estimation. CO-3 Gradient descent for linear models. Derivatives of MSE and cross entropy loss functions. CO-4 The problem of overfitting. CO-5 Regularization for linear models.
41.	19B1WCI734	Object-Oriented Technologies using Java	CO1 Understanding the features and concepts of Object-Oriented Programming using Java. CO2 Defining classes, objects, constructors, methods in Java. CO3 Inheritance, Interfaces, and Polymorphism. CO4 Packages, Exception Handling, and Multithreading

			CO1 Understanding the features and concepts of Object-Oriented Programming using Java.
42.	19B1WCI774	Object-Oriented Technologies using Java Lab	CO2 Defining classes, objects, constructors, methods in Java.
			CO3 Inheritance, Interfaces, and Polymorphism.
			CO4 Packages, Exception Handling, and Multithreading.
43.	19B1WCI740	Introduction to Statistical learning	CO-1 Understanding statistical learning and, analyzing regression and classification.
			CO-2 Analyzing resampling, model selection and regularization.
			CO-3 Analyzing non-linear and tree based regression.
			CO-4 Analyzing Support Vector Machines and unsupervised learning
44.	18B19CI791	MAJOR PROJECT-I	CO 1: To Plan/design and code, dependable software/hardware.
			CO 2: To act as catalyst in transferring the Computer Science and Engineering and Information Technology knowledge to field usage for the socio-economic development of the society.

Semester: VII

SNo	Course Codes	Course Name	Course Outcomes
1.	10B19CI891	Project Part 2	CO-1 To Plan/design and code, dependable software/hardware. CO-2 To act as catalyst in transferring the Computer Science and Engineering and Information Technology knowledge to field usage for the socio-economic development of the society.
2.	10M11CI211	Advanced Algorithms	CO1: Analyze the asymptotic performance of algorithms. CO2: Write rigorous correction proof s of algorithms. CO3: Demonstrate a familiarity with major algorithms and data structure. CO4: Apply important algorithmic design paradigms and method of analysis.
3.	10M11CI212	Advanced Operating Systems	CO1: Able to implement a simple distributed application using a message based protocol. CO2: To model connection-oriented and connectionless communication in a 2 tier Client Server architecture. CO3: Able to distinguish the five main failure types in a Distributed System and specify algorithms for achieving fault tolerance and error recovery within such a system. CO4: Implement a remote object based system to demonstrate parameter passing and code migration in a Distributed System. CO5: To learn to achieve synchronisation among a group of processes in a distributed system.

			CO6: To understand algorithms for determining global state, electing a co-ordinator for a group of communicating processes and implementing mutual exclusion in a Distributed System. CO7: To be able to differentiate between client centric and data centric consistency models and describe protocols for implementing consistency models and updating replicas in a Distributed System.
4.	10M11CI213	Advanced Software Engg	CO1: Review and understand the software Process. CO2: Emphasise improvement in software Quality. CO3: Review and understand various software architecture blueprints. CO4: Practice software reuse and adopt common design patterns. CO5: Understand component based software development. CO6: Understand test driven software development the agile way. CO7: Study software metrics and cost estimation techniques CO8: Learn Professional software development tools and techniques.
5.	11B1WCI834	Parallel Processing	CO-1: Describe different parallel processing architectures based on relationships between processing elements, instruction sequence, memory and interconnected network. CO-2: Identify algorithms, which require parallelization as part of system design or performance enhancement. CO-3. Classify shared and distributed memory parallel systems according to their properties and usage models CO-4: Design and develop parallel algorithms for shared and distributed memory models. CO-5: Evaluate the performance of parallel algorithms designed based on shared and distributed memory models as well as against serial based algorithm designs.
6.	11B1WCI835	Storage Networks	CO-1: Understand various components and protocols used for creating storage networks. CO-2: Learn about various technologies existing for storage networks. CO-3: Explore Storage virtualization. CO-4 Analyse techniques for management of storage networks. About Data Centers and Distributed Storage Networks.
7.	11B1WCI836	Network Management(OE)	CO1: Become familiar with the Network Management Standards. CO2: Understand the SNMP protocols. CO3: Understand how large-scale Network Management Systems operate. CO4: Understand how large-scale Network Management Systems are configured.

			CO5: Advanced network Management Tools and Systems. CO6: Web Based Network Management Systems.
8.	12B1WCI831	Cloud Computing	CO1: To learn the basic concepts, applications and terminology of cloud computing. CO2: To understand different enabling technologies for Cloud computing environment. CO3: To design Cloud computing data-center for effective utilization of available resources. CO4: To study different managers related to Cloud computing services. CO5: To understand different case studies of Cloud computing and its advance topics.
9.	13B1WCI731	ARM Based Embedded System Design	CO1: To learn the basic concepts of Computer Architecture. CO2: To learn ARM Software Development Tools and ARM Architecture Fundamentals. CO3: Understanding ARM Assembly Language Programming Basics. CO4: To learn Programming Techniques for ARM.
	13B1WCI831	Service Oriented Architecture	CO-1: To gain understanding of the basic principles of service orientation. CO-2: To learn service oriented analysis techniques. CO-3: Learn technology underlying the service design.
10.	14M1WCI431	Parallel Programming Techniques	CO-1: Describe different parallel processing architectures based on relationships between processing elements, instruction sequence, memory and interconnected network. CO-2: Identify algorithms, which require parallelization as part of system design or performance enhancement. CO-3: Classify shared and distributed memory parallel systems according to their properties and usage models. CO-4: Design and develop parallel algorithms for shared and distributed memory models. CO-5: Evaluate the performance of parallel algorithms designed based on shared and distributed memory models as well as against serial based algorithm designs.
11.	15B1WCI831	Wireless Sensor Networks- Protocols and Applications(OE)	CO-1 Architect sensor networks for various application setups. CO-2 Explore the design space and conduct trade-off analysis between performance and resources. CO-3 Assess coverage and conduct node deployment planning. CO-3 Devise appropriate data dissemination protocols and model links cost. CO-4 Determine suitable medium access protocols and radio hardware. CO-5 Prototype sensor networks using commercial components. CO-6 Provision quality of service, fault-tolerance, security

				and other dependability requirements while coping with resource constraints. CO-7 Evaluate the performance of sensor networks and identify bottlenecks. CO-8 Explore Security, routing Protocol and information Fusion in WSN.
12.	15B1WCI832		Internet of Things Architecture	CO-1: Understand the basics of IoT, The various IoT protocols and applications of various IoT technologies. CO-2: Learn the basics of Contiki OS and standardized protocols for IPv6 Low power networking. CO-3 Understand the IEEE 802.15.4 standard and 6 LoWPAN technology. CO-4 Understand the RPL protocol and to understand the various routing Issues in IoT. CO-5 Analyse the security issues, data collection and distributed computing.
13.	15B1WCI833		Big Data Analytics	CO-1 Understand the need for Big Data Analytics. CO-2 Master the concepts of large scale file systems and map reduce framework. CO-3 Master the concepts of mining data streams. CO-4 Master the concepts of Link analysis and frequent item sets discovery from Big data. CO-5 Master the concepts of clustering for streams and parallelism.
14.	17B1WCI811		Mobile and Distributed Computing	CO-1: Introduce enabling technologies of pervasive computing. Co-2 Acquire solid knowledge on mobile networks and mobile computing. CO-3: Develop applications that are mobile-device specific and demonstrate current practice in mobile computing contexts. CO-4: Understand the concepts of Ashco and wireless sensor networks. CO-5: Understand the concepts of Mobile IP. CO-6: To be able to analyse the performance of different handoff, roaming, and location update algorithms for cellular networks. CO-7: Awareness of professional and ethical issues, in particular those relating to security and privacy of user data and user behaviour. CO-8 Describe the possible future of mobile computing technologies and applications.
15.	17B1WCI812		Computer Games	CO-1: Discuss games and game design. It includes the history and philosophy of games, the game production process. CO-2: Discuss introduction to the electronic game design and development careers. CO-3: Discover what the components of games are, and what parts of games are influenced by their design. CO-4: Learn several ways to approach the design of a

			game, and processes and best practices for prototyping, play testing and balancing a game after it has been designed.
16.	17B1WCI814	Design and Analysis of Real-World Algorithms	CO-1: To acquire knowledge of Machine learning algorithms. CO-2: To gain knowledge on the notions Cryptographic algorithms. CO-3: To have hands on Network Algorithms and their applications in problem solving. CO-4: To have hands on skills to evaluate Compression algorithms and their applications and implementations in day to day problem solving. CO-5: To evaluate different kinds of Geometry and graphics algorithms and their applications and implementations in simulations. CO-6: To gain understanding of Algorithms for Big Data : representation and algorithms.
17.	18B1WCCI84 2	R Programming and Application	CO1: To identify the usages of available R packages and associated Open Source software to meet different scientific objectives. CO2: To understand how to programming in R, reading data into R, accessing R packages. CO3: Able to write R functions, debugging, profiling R code, and organizing and commenting R code. CO4: To design and write efficient programs using R to perform routine and specialized data manipulation/management and analysis tasks. CO5: To do data analysis using R for real life applications.
18.	18B1WCI831	Deep Learning	CO1: Variability models (deformation model, stochastic model). CO2: Properties of CNN representations: invertibility, stability, invariance. CO3: Covariance/invariance: capsules and related models. CO4: Other tasks: localization, regression. CO5: Dynamical systems: RNNs. CO6: Autoencoders (standard, denoising, contractive, etc etc) CO7: Maximum Entropy Distributions. CO8: Non-convex optimization for deep networks. CO9: Stochastic Optimization.
19.	18B1WCI832	Machine Learning Algo	CO1: To learn the basic concepts and terminology in machine learning. CO2: To learn about the definition of learning systems, their goals and applications in machine learning. CO3: To understand concepts associated with classification and experimental evaluation of classification algorithms.

			CO4: To learn concepts associated with decision trees and experimental evaluation of classification algorithms. CO5: To learn about instance-based learning, clustering and unsupervised learning.
20.	18B1WCI836	Block Chain Technologies	CO1: Explain how bitcoin works, from when a transaction is created to when it is considered part of the blockchain. CO2: Thoroughly explain private and public keys as well as addresses and how exactly they are constructed and used. CO3: Expose students to the different kinds of forking and explain the Bitcoin's network mechanisms for maintaining and upgrading. CO4: Decompose a blockchain system's fundamental components, how they fit together and examine a modular blockchain system in more detail. CO5: Detailed understanding of naïve Attacks and Trustless Networks of block chain. CO6: Provide a thorough understanding of smart contracts, their technical capabilities, practical applications, limitations and security constraints they operate within. CO7: Explain to students both fundamental and implied differences between Ethereum and Bitcoin protocol by covering historical, conceptual and architectural distinctions.
21.	18B1WCI838	Secure Web Application Development	CO1: Veracode offers multiple analysis techniques for identifying vulnerabilities. CO2: Web Application Discovery and Monitoring tools perform scans to discover and inventory all external web applications, even the apps that organizations didn't know were still running. CO3: Static Application Security Testing (SAST) scans applications from the "inside out", reviewing static code for common vulnerabilities such as cross-site scripting and SQL injection. CO4: Dynamic Application Security Testing (DAST) searches for flaws in software already in production and does not require access to source code. CO5: Manual penetration testing enables development teams to find vulnerabilities such as authorization issues and business logic flaws that can only be discovered with the help of a skilled penetration tester.
22.	18B1WCI839	Natural Language Processing Tech	CO1: Understand Natural Language Processing (Understanding). CO2: Probabilistic model of defining language and techniques.(Application) CO3: Applying Hidden Markov model and Speech Recognition.(Application)

			CO4: Application of context free grammar and language parsing.(Application) Implement probabilistic and language parsing.(Application) CO5: Differentiation of semantic and discourse in terms of NLP.(Analyse)
23.	18BIWCI832	Social and Information Network Analysis	CO-1. Understand what constitutes a social network. CO-2. Represent networks in graph-theoretic language. CO-3. Design effective and reliable network research projects. CO-4. Configure network and attribute data for standard software packages. CO-5. Comprehend the reasons for using, and principles of, permutation tests. CO-6. Apply centrality measures appropriately.
24.	19BIWCI831	Ethics and Information Technology	CO-1 Demonstrate knowledge of current models of information and computer ethics. CO-2 Apply ethical theories to interpret personal and group behavior when using a variety of information technology tools. CO-3 Evaluate the nature of ethical choices made by self and others when serving various roles that expose social and multicultural differences. CO-4 Construct written arguments in a variety of formats on the evolving nature of ethical norms relating to new technologies.
25.	19BIWCI832	Probabilistic Graphical Models	CO-1 Learn introductory concepts in probabilistic graphical models. CO-2 Able to model problems using graphical models. CO-3 Able to model problems using design inference algorithms. CO-4 Analyse the structure of the graphical model from data.
26.	19BIWCI833	Information Modelling	CO-1 Understand the basics of information modelling application and their usages in different areas. CO-2 To be familiar with the basics of various modelling techniques. CO-3 Ability to apply modelling techniques on real-life scenarios to solve them easily. CO-4 Ability to apply information modelling with Linked data.
27.	19BIWCI835	Cloud Computing Security	CO-1 To understand the use of cloud computing. CO-2 A different technique to secure the cloud. CO-3 Different types of attacks on the cloud. CO-4 Identify the known threats, risks, vulnerabilities.
28.	19BIWCI837	Reinforcement Learning	CO-1 Understanding the basic concepts of Reinforcement Learning. CO-2 Understanding Probability Primer Techniques in detail. CO-3 Learning & implementing various methods & techniques such as TD methods, Monte Carlo, Markov

				Decision Process, etc. CO-4 Learning & understanding policy gradients along with its advantages and disadvantages.
29.	19BIWCI839	Foundations of Blockchain		CO-1 Explain how bitcoin works, from when a transaction is created to when it is considered part of the blockchain. CO-2 Thoroughly explain private and public keys as well as addresses and how exactly they are constructed and used. CO-3 Expose students to the different kinds of forking and explain the Bitcoin's network mechanisms for maintaining and upgrading. CO-4 Decompose a blockchain system's fundamental components, how they fit together and examine a modular blockchain system in more detail. CO-5 Detailed understanding of naïve Attacks and Trustless Networks of block chain. CO-6 Provide a thorough understanding of smart contracts, their technical capabilities, practical applications, limitations and security constraints they operate within. CO-7 Explain to students both fundamental and implied differences between Ethereum and Bitcoin protocol by covering historical, conceptual and architectural distinctions.
30.	20B1WCI772	From Graph to Knowledge Graph Lab		CO-1 Build a knowledge graph from documents. CO-2 Building a knowledge graph from a text. CO-3 Build a Small Knowledge Graph. CO-4 Understand the Graph Databases. CO-5 Familiarity with graph computing Frameworks. CO-6 Able to Visualization of graphs.
31.	20BIWCI732	From Graph to Knowledge Graph		CO-1 Learn graph representations. CO-2 Utilize fundamentals to build knowledge graphs. CO-3 Understand the algorithms and techniques for crawling web sites, structured data extraction, and information extraction from unstructured text. CO-4 Understand how to apply the tools and infrastructure to build and query knowledge graphs.
32.	18B19CI891	MAJOR PROJECT-II		CO 1: To Plan/design and code, dependable software/hardware. CO 2: To act as catalyst in transferring the Computer Science and Engineering and Information Technology knowledge to field usage for the socio-economic development of the society.