**Department of Computer Science**

**Course Offered**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Name of Course** | **Subject** | **Level** |
| **1.** | **M.Sc** | **Computer Science** | **PG** |
| **2.** | **B.Sc** | **Physics, Chemistry and Mathematics** | **UG** |
| **3.** | **Certificate Course** | 1. **Programming in C**
2. **Programming in C++**
3. **MS Office**
4. **MS Excel**
5. **HTML**
 | **UG/PG** |

**Program Outcomes (POs) for B. Sc Programme After completing the graduate studies in science, students will be able to**

**PO1**: Demonstrate proficient application of theoretical knowledge and practical skills gained through the graduate program in their respective field of work.

**PO2**: Evaluate and analyze both simple and complex real-life situations, utilizing scientific concepts to provide effective solutions.

**PO3**: Acquire skill sets that contribute to the overall development and progress of the nation.

 **PO4**: Acquire industry-specific skills to remain competitive in a rapidly evolving global environment.

**PO5**: Apply acquired skills to independently design and conduct experiments, interpret data, develop hypotheses, and foster a curious and innovative mindset towards research.

 **PO6**: Recognize the societal and environmental impact of scientific solutions and demonstrate an understanding of the importance of sustainable development.

 **PO7**: Augment learning with skill enhancement courses and value-added programs to provide students with a competitive advantage, particularly in the global context.

**PO8**: Instill ethical, moral, and social values in personal and social life, cultivating individuals who are refined and conscientious.

 **PO9**: Foster holistic education that nurtures the comprehensive development of students, encompassing academic, personal, and social aspects.

**DEPARTMENT OF COMPUTER SCIENCE COURSE OUTCOME**

**Paper – I : Programming in C**

**CO1** 1. Students will gain the skills to design and implement algorithms. They will learn how to break down a problem into smaller subproblems and use control structures to create efficient and logical solutions.

2. Students will develop skills in algorithmic thinking and problem-solving, enabling them to design efficient algorithms and implement them.

**CO2** 1. Students will grasp the fundamental concepts of programming, such as variables, data types, operators , control structures (e.g., loops and conditionals), functions, and arrays. They will learn how to write simple programs using these elements.

 2. Students will become familiar with the syntax and semantics of the C programming language, including the proper use of statements, expressions, and declarations.

 3. Students will develop the skills to write C programs to solve various computational problems. They will learn how to write code that is clear, efficient, and follows best programming practices.

 4. Students will learn how to identify and fix errors (bugs) in their programs using debugging techniques and strategies. They will gain proficiency in troubleshooting common programming issues.

 5. Students will gain knowledge about various standard libraries and functions available in C, such as the standard input/output library (stdio.h), string manipulation functions (string.h), mathematical functions (math.h), and others.

 6. Students will gain practical experience in writing, compiling, and executing C programs using integrated development environments (IDEs)

**CO3** 1. Students will grasp the concept of arrays as a collection of elements of the same data type stored in contiguous memory locations. They will learn how to declare and initialize arrays, as well as access and manipulate individual array elements.

 2. Students will be introduced to coding best practices specific to string handling in C. This includes guidelines for string manipulation, error handling, memory management, code readability, and efficient algorithms for string operations.

 3. Introduce guidelines for function naming conventions, parameter naming, code readability, modularity, and documentation to enhance code quality and maintainability.

 4. Students will be introduced to coding best practices specific to storage classes in C. This includes guidelines for variable declaration, choosing appropriate storage classes, and understanding the impact of storage classes on program performance and memory usage.

**CO4** 1. Students will gain a solid understanding of how to declare, initialize, and use pointers. They will learn about pointer arithmetic and the role of pointers in accessing and manipulating data.

 2. Students will grasp the concept of structures as a composite data type that allows the grouping of related data items under a single name. They will learn how to define and declare structures, as well as access and manipulate their individual members.

 3. Students will learn about the differences between unions and structures in terms of memory allocation and member access. They will understand that unions store only one member at a time, whereas structures store all members simultaneously.

 4. Students will learn how to read from and write to files using the file I/O functions available in C. They will understand concepts like file pointers, opening and closing files, and performing various operations on files.

 5. Students will grasp the concept of command line arguments and their role in passing inputs to C programs during runtime. They will learn how command line arguments can be used to provide flexibility and user interaction.

 6. Students will learn how to read from and write to files in C. They will understand concepts like file pointers, opening and closing files, reading and writing data, and error handling related to file operations.

**Paper – II : Introduction to Information Technology**

**CO1** Students will understand the basic principles of Information Technology like computerbased system and component to meet desired needs.

**CO2** Understand fundamentals of the data/signal transmission over communication media

 **CO3** Understand the transmission media and their standards to practice different protection schemes at individual and team level.

 **CO4** Explore the concept of network topology, and different ways of communication between PCs using Wi-Fi, Bluetooth and Infrared devices. And understand the architecture of peer-to-peer and client/server.

**Semester : II**

**Paper-I : Object Oriented Programming Using ‘C++’**

**CO1** 1. Students will grasp the core concepts of OOP, including encapsulation, inheritance, and polymorphism. They will learn how these concepts contribute to code organization, reusability, and maintainability.

2. Students will learn how to define classes, create objects from those classes, and understand the relationship between classes and objects in OOP. They will gain insights into class structure, member variables, and member functions.

3. Students will grasp the concept of static data members in C++ and their behavior. They will learn that static data members belong to the class itself rather than individual objects, and they are shared among all objects of the class.

4. Students will grasp the concept of access specifiers in C++ and their role in controlling the visibility and accessibility of class members. They will learn about three access specifiers: public, private, and protected.

**CO2** 1. Students will grasp the concept of constructors in C++ and their role in initializing objects of a class. They will learn about default constructors, parameterized constructors, copy constructors, and their syntax and usage.

 2. Students will grasp the concept of operator overloading in C++ and its role in providing customized behavior for operators when working with user-defined types and classes.

3. Students will gain insights into how operator overloading can be used to create expressive and intuitive interfaces for classes. They will understand how operator overloading can enhance the object-oriented design and usability of user-defined types.

**CO3** Students will be able to :

1. Demonstrate a clear understanding of dynamic memory allocation in C++.

2. Use new and delete operators to dynamically create and destroy objects during program execution.

3. Implement dynamic object creation and destruction using constructors and destructors.

4. Manipulate dynamic objects using pointers and understand the concept of pointer arithmetic.

5. Understand the concept of inheritance and its significance in object-oriented programming.

6. Analyze and compare different inheritance approaches (e.g., single inheritance, multiple inheritance, virtual inheritance) and select appropriate techniques based on specific design requirements.

 **CO4** Students will be able to : 1. Understand the concept of virtual functions in C++ and their significance in achieving polymorphic behaviour.

2. Apply the concept of function overriding to redefine base class functions in derived classes, considering access specifiers and return types.

3. Utilize virtual destructors to ensure proper destruction of derived class objects through base class pointers.

4. Understand the concept of exception handling in C++ and its importance in managing program errors and ensuring robustness.

5. Utilize the try-catch block to handle exceptions and prevent program termination due to errors.

**Semester : II**

**Paper-II : System Analysis and Design**

**CO1** Define the system development life cycle. Conducts research on existing systems and develop plans for the new system.

 **CO2** Make the feasibility study about the system (Technical risks and technical possibilities).

**CO3** Scheduling with using GANTT and PERT techniques. Evaluates the economic selfsufficiency whether to install the system.

**CO4** Carry out the system analysis. Identifying the problems in the system. Determine the cause of the problem in the system. Find a solution of the problem in the system.

**Semester : III**

**Paper- I : Data structure**

**CO1** Able to write the algorithms and implementing the algorithms based on Single linked List and Double Linked list.

**CO2** Able to write algorithms and implemented based on Stack and also understood the concept and Application of stack

**CO3** Able to write the algorithms and implementing the algorithms based on Queue and sorting Techniques.

 **CO4** Able to understand the concept of Binary tree and Graphs and Traversing methods of Trees and Graphs.

**Semester : III**

**Paper- II : Operating System**

**CO1** Able to understand structure of OS, Process management, Schedulers and different Scheduling algorithms.

**CO2** Able to understand the concept related to Deadlock and Starvation.

**CO3** Able to understand the concept related to memory management, partition Description table, Paging and Segmentation

 **CO4** Able to understand the concept of I/O management ,File management and Protection mechanism

**Semester : IV**

**Paper- I : Java Programming**

**CO1 Students will be able to :**

1.Understand timeline, features of Java programming language.

2. Understand the concept of variables and their role in storing and manipulating data in Java.

3. Differentiate between different data types in Java, such as primitive types (int, double, boolean, etc.) and reference types (classes, arrays, etc.).

**CO2 Students will be able to :**

1. Understand the concept of classes and objects in object-oriented programming (OOP) and their significance in Java.

2. Implement access specifies to enforce encapsulation and data hiding principles in class design.

3. Develop well-structured, efficient, and scalable Java programs that effectively utilize method overloading to enhance code flexibility and reusability.

4. Understand the structure and organization of the Java Class Library, including the core packages (java.lang, java.util, etc.) and their respective functionalities.

5. Understand the concepts of decision making and conditional statements in Java.

6. Develop well-structured, efficient, and maintainable Java programs that effectively utilize arrays for data storage and manipulation.

7. Understand the concept of inheritance in object-oriented programming and its significance in code reuse and hierarchy establishment.

8. Develop well-structured, efficient, and scalable Java programs that effectively utilize inheritance for code reuse and maintainable design.

9. Develop well-structured, efficient, and scalable Java programs that effectively utilize interfaces for abstraction, code reusability, and maintainable design.

**CO3 Students will be able to :**

1. Understand the package naming conventions and guidelines for creating and naming packages.

2. Implement basic applet functionality, including rendering graphics, handling user input, and responding to events.

3. Develop well-structured, efficient, and scalable Java programs that effectively utilize threads for concurrent programming and multitasking.

 4. Understand the concept of exceptions and errors in Java and their role in handling exceptional and error conditions.

**CO4 Students will be able to :**

1. Understand the event-driven programming model and utilize event listeners and handlers to respond to user actions and events.

2. Design and create GUI components using AWT, such as buttons, labels, text fields, checkboxes, radio buttons, and list boxes.

3. Implement event handling for user actions, such as button clicks, menu selections, and keyboard events.

4. Develop well-structured, efficient, and user-friendly GUI applications in Java using AWT that meet specific requirements and usability standards.

**Semester- IV**

**Paper – II : Linux Operating System**

 **CO1** Understand the basic commands of Linux operating system.

 **CO2** Create file systems and directories and managing hardware.

 **CO3** Working on files (Sharing files with others, granting and revoking file access). Managing the user’s account. Understand the processes background and fore ground by process and signals system calls.

 **CO4** Create shared memory segments, pipes, message queues and can exercise inter process communication. Working with different graphical user interfaces (KDE & GNOME)

**Paper – I : Visual Basic Programming.**

**Semester- V**

**CO1**

1. Able to program in VB using controls

2. Understand to handle data types, loops and control structures

**CO2** 1. Able to handle homogeneous data

2. Apply code reusability with procedures, functions and modules

**CO3** 1. Design interface using Menus

2. Apply DAO to handle database

**CO4** 1. Apply ADO to handle database

2. Able to handle errors in program

**Paper – II : Data Base Management System.**

**Semester- V**

**Course Outcomes(COs)**

**CO1** Concept related to DBMS, Comparative differences with traditional file system and Non procedural concept and different Data models

**CO2** Able to construct Entity Relationship diagram, understanding the concept of strong and Weak Entity sets.

**CO3** Able to perform different operations on Database and concept related to Aggregate functions.

**CO4** Ability to perform Normalized the database using different normal forms.

**Paper – I : Compiler Construction**

**Semester- VI**

**Course Outcomes(COs)**

**CO1** Able to understand the concept of Compilers and Translators and different phases of Compiler.

**CO2** Able to understand the definitions of programming languages structure operations and storage management.

**CO3** Able to understand the role of Lexical Analyzer, syntax Analyzer, Context free Grammer, Ambiguous Grammar.

**CO4** Able to understand the concept of different Parsing techniques and DAG representation.

**Paper – II : SQL and PL/SQL**

**Semester- VI**

**Course Outcomes(COs)**

**CO1** Able to understand creating table, constraints, different Data types, functions, operations, different Database Sublanguages

**CO2**  Able to understand creating views ,PL/SQL programming Data types, Iterative and conditional statements and problem based on PL/SQL

**CO3** Able to understand Exceptions, writing cursors and types creating procedures and Examples on procedures

**CO4** Able to understand the concept of Functions, Purity levels in functions, Triggers, Types of Triggers, Enabling disabling Triggers and problem based on Triggers.

**Course Outcome of M.Sc (Computer Science)**

M.Sc Computer Science is a Postgraduate course which is affiliated to the Rashtrasant Tukdoji Maharaj Nagpur University. The course comprises four semesters, each semester has four subjects and two practicals. The final semester includes project work instead of a practical exam where the students get a chance to showcase the knowledge they have gathered in their course tenure.


##### M.Sc Computer Science Semester I

|  |
| --- |
| **Paper - I (Code : 1T1) Discrete Mathematical Structure** |
| **CO 1** | Understand the basic concepts of set theory, logic, functions, relations and graph theory. |
| **CO 2** | Use mathematical proofs to demonstrate the correctness of algorithms and other mathematical statements. |
| **CO 3** | Design and implement algorithms to solve problems in discrete mathematics. |
| **CO 4** | Understand the basic concepts of semigroups and groups. |

|  |
| --- |
| **Paper - II (Code : 1T2) Programming in Java** |
| **CO 1** | Understand the basic concepts of object-oriented programming |
| **CO 2** | Implement Java programs using the basic syntax and constructs |
| **CO 3** | Understand the concepts of JDBC, networking, and RMI. |
| **CO 4** | Understand the concepts of servlets, JSP, and JavaBeans. |

|  |
| --- |
| **Paper - III (Code : 1T3) Digital Electronics and Microprocessor** |
| **CO 1** | Understand the basic concepts of number systems and data representation. |
| **CO 2** | Be able to use Boolean algebra to simplify logic expressions. |
| **CO 3** | Be able to design and analyze shift registers. |
| **CO 4** | Understand the architecture of the 8086 microprocessor. |

|  |
| --- |
| **Paper - IV (Code : 1T4) Advanced DBMS and Administration** |
| **CO 1** | Understand the basic concepts of relational database design, including functional dependencies, normalization, and query processing. |
| **CO 2** | Be able to design and implement transaction processing systems. |
| **CO 3** | Understand the basic concepts of Oracle database architecture and administration, including data dictionary views, standard packages, and managing rollback segments. |
| **CO 4** | Be able to tune Oracle databases for performance. |

|  |
| --- |
| **Practical - I (Code : 1P1)** |
| **CO 1** | Implement Java programs using the basic syntax and constructs |
| **CO 2** | Apply Java to solve real-world problems |

|  |
| --- |
| **Practical - II (Code : 1P2)** |
| **CO 1** | Program a microprocessor using simulators |

**Semester II**

|  |
| --- |
| **Paper - I (Code : 2T1) Windows Programming using VC++** |
| **CO 1** | Understand the basic concepts of Windows programming, such as the Windows API, GDI, and MFC. |
| **CO 2** | Be able to develop graphical user interfaces using GDI and MFC. |
| **CO 3** | Be able to think logically and solve problems using Windows programming. |
| **CO 4** | Be able to apply Windows programming skills to real-world problems. |

|  |
| --- |
| **Paper - II (Code : 2T2) Theory of Computation and Compiler Construction** |
| **CO 1** | Understand the basic concepts of compiler construction, including lexical analysis, parsing, code generation, and optimization. |
| **CO 2** | Be able to design and implement finite-state machines. |
| **CO 3** | Be able to think critically about the design and implementation of compilers. |
| **CO 4** | Be able to apply the theory of computation and compiler construction to solve real- world problems. |

|  |
| --- |
| **Paper - III (Code : 2T3) Computer Architecture and Organization** |
| **CO 1** | Understand the basic concepts of computer organization, such as the processor, memory, and I/O devices. |
| **CO 2** | Understand the design of the control unit, including the data path and control path. |
| **CO 3** | Understand the different storage technologies used in computer systems. |
| **CO 4** | Understand the concept of performance evaluation in computer systems. |

|  |
| --- |
| **Paper - IV (Code : 2T4) Computer Graphics** |
| **CO 1** | Understand the basic concepts of computer graphics, such as 2D and 3D graphics, geometric transformations, and rasterization. |
| **CO 2** | Understand the different line drawing algorithms, such as DDA and Bresenham's. |
| **CO 3** | Understand the basic transformation operations, such as translation, rotation, and scaling. |
| **CO 4** | Understand the different color models, such as XYZ, RGB, YIQ, CMY, and HSV. |

|  |
| --- |
| **Practical - I (Code : 2P1)** |
| **CO 1** | Understand the basic concepts of Visual C++, such as variables, data types, operators, and control flow statements. |
| **CO 2** | Understand the basic concepts of object-oriented programming, such as classes, objects, and inheritance. |
| **CO 3** | Be able to write simple Visual C++ programs and be able to use the Visual Studio IDE. |

|  |
| --- |
| **Practical - II (Code : 2P2)** |
| **CO 1** | Be able to use C++ graphics libraries to create and manipulate graphics images. |
| **CO 2** | Be able to apply computer graphics concepts to real-world problems. |

##### Semester III

|  |
| --- |
| **Paper - I (Code : 3T1) Data Communication and Networks** |
| **CO 1** | Understand the basic concepts of data communication and networking, such as network structure, architectures, and services. |
| **CO 2** | Understand the different types of layers, protocols and how they are used. |
| **CO 3** | Understand the basic concepts of network security, such as security vulnerabilities and threats, and classification of security services. |
| **CO 4** | Understand the design principles of firewalls and how they are used to filter packets and control access to computer networks. |

|  |
| --- |
| **Paper - II (Code : 3T2) Software Engineering** |
| **CO 1** | Understand the basic concepts of software engineering, such as the software development life cycle, software quality, and software testing. |
| **CO 2** | Be able to apply the software development life cycle to a software project. |
| **CO 3** | Be able to work collaboratively on software engineering projects. |
| **CO 4** | Be able to use software testing techniques to find defects in software. |

|  |
| --- |
| **Paper - III (Core Elective 1) (Code : 3T3) CE1-1 Neural Network** |
| **CO 1** | Understand the basic concepts of neural networks, such as artificial neurons, layers, and activation functions. |
| **CO 2** | Understand the different learning algorithms used in neural networks, such as backpropagation and stochastic gradient descent. |
| **CO 3** | Be able to implement neural networks in a programming language, such as Python or Java. |
| **CO 4** | Be able to apply neural network concepts to real-world problems. |

|  |
| --- |
| **Paper - III (Core Elective 1) (Code : 3T3) CE1-2 Multimedia Technologies** |
| **CO 1** | Understand the basic concepts of multimedia, such as images, audio, and video. |
| **CO 2** | Understand the basics of ActionScript, including object-oriented programming concepts, data types, and classes. |
| **CO 3** | Understand the principles of multimedia data compression, including lossless and lossy compression algorithms. |
| **CO 4** | Understand the basic concepts of audio compression, including quantization and wavelet-based coding. |

|  |
| --- |
| **Paper - III (Core Elective 1) (Code : 3T3) CE1-3 ASP.NET** |
| **CO 1** | Understand the basics of ASP.NET, including the event-driven programming model, the HTTP protocol, and the structure of ASP.NET pages. |
| **CO 2** | Understand the different types of server controls in ASP.NET, including HTML controls, web controls, and validation controls. |
| **CO 3** | Understand how to connect to data sources in ASP.NET. |
| **CO 4** | Understand how to manage state in ASP.NET, including application state, session state, and view state. |

|  |
| --- |
| **Paper - IV (Foundation Course 1) (Code : 3T4) FC1 Operating System Concepts** |
| **CO 1** | Understand the different operating system services, such as process management, memory management, and file management. |
| **CO 2** | Understand the different operating system scheduling algorithms, such as round robin, priority scheduling, and shortest job first scheduling. |
| **CO 3** | Be able to design and implement operating system features. |
| **CO 4** | Be able to communicate effectively about operating system concepts. |

|  |
| --- |
| **Paper - IV (Core(Discipline Centric)1) (Code : 3T4) CDC1 Mobile Computing** |
| **CO 1** | Understand the basic concepts of mobile computing, such as mobile devices, mobile networks, and mobile applications. |
| **CO 2** | Understand the different types of mobile networks, such as GSM, GPRS, EDGE, 3G, and 4G. |
| **CO 3** | Be able to communicate effectively about mobile computing concepts. |
| **CO 4** | Understand the challenges of mobile computing, such as limited resources, intermittent connectivity, and security. |

|  |
| --- |
| **Practical - I (Code : 3P1)** |
| **CO 1** | Be able to write C++ code to create and use sockets. |
| **CO 2** | Be able to implement different networking protocols in C++. |

|  |
| --- |
| **Practical - II (Code : 3P2)** |
| **CO 1** | Understand the ASP.NET component model and the ASP.NET provider model. |
| **CO 2** | Understand the anatomy of an ASP.NET page, including the Page class and the Page lifecycle. |
| **CO 3** | Understand the different types of server controls in ASP.NET, including HTML controls, web controls, and validation controls. |

**Semester IV**

|  |
| --- |
| **Paper - I (Code : 4T1) Data Mining** |
| **CO 1** | Understand the different types of data mining tasks, such as classification, clustering, association rule mining, and sequential pattern mining. |
| **CO 2** | Understand the different types of data mining algorithms, such as decision trees, support vector machines, and k-means clustering. |
| **CO 3** | Be able to evaluate the performance of data mining algorithms. |
| **CO 4** | Be able to think critically about data mining problems and come up with innovative solutions. |

|  |
| --- |
| **Paper - II (Code : 4T2) Artificial Intelligence & Expert System** |
| **CO 1** | Understand the different types of artificial intelligence systems, such as rule-based systems, case-based reasoning systems, and neural networks. |
| **CO 2** | Understand the different applications of artificial intelligence, such as natural language processing, computer vision, and robotics. |
| **CO 3** | Be able to design and implement artificial intelligence systems. |
| **CO 4** | Be able to think critically about artificial intelligence problems. |

|  |
| --- |
| **Paper - III (Core Elective 2) (Code : 4T3) CE2-1 Design and Analysis of Algorithm** |
| **CO 1** | Understand the different types of algorithms, such as greedy algorithms, divide-and- conquer algorithms, and dynamic programming algorithms. |
| **CO 2** | Understand the different types of problem-solving techniques, such as backtracking and branch-and-bound. |
| **CO 3** | Be able to solve problems using different problem-solving techniques. |
| **CO 4** | Be able to think critically about algorithm design problems. |

|  |
| --- |
| **Paper - III (Core Elective 2) (Code : 4T3) CE2-2 Embedded System** |
| **CO 1** | Understand the basic concepts of embedded systems, such as hardware, software, and firmware. |
| **CO 2** | Understand the different types of embedded systems, such as real-time systems, embedded controllers, and intelligent devices. |
| **CO 3** | Be able to design and implement embedded systems. |
| **CO 4** | Be able to think critically about embedded systems problems. |

|  |
| --- |
| **Paper - III (Core Elective 2) (Code : 4T3) CE2-3 Pattern Recognition** |
| **CO 1** | Understand the different types of pattern recognition algorithms, such as decision trees, neural networks, and support vector machines. |
| **CO 2** | Be able to apply pattern recognition techniques to solve problems. |
| **CO 3** | Understand the different types of clustering algorithms and how they are used to group patterns. |
| **CO 4** | Be able to evaluate the performance of pattern recognition systems and identify areas for improvement. |

|  |
| --- |
| **Paper - IV (Foundation Course 2) (Code : 4T4) FC2 Advances in Information Technology** |
| **CO 1** | Understand the different types of software and their purposes. |
| **CO 2** | Understand the different types of programming languages and their strengths and weaknesses. |
| **CO 3** | Understand the different types of networks and network topologies and their advantages and disadvantages. |
| **CO 4** | Understand the concept of e-commerce and the internet and how they are used in business. |

|  |
| --- |
| **Paper - IV (Core(Discipline Centric)2) (Code : 4T4) CDC2 Parallel Computing** |
| **CO 1** | Understand the basic concepts of parallel computing, such as parallelism, scalability, and performance. |
| **CO 2** | Understand the principles of parallel algorithm design, such as decomposition techniques, mapping techniques, and communication operations. |
| **CO 3** | Understand the analytical modeling of parallel programs, such as performance metrics and scalability metrics. |
| **CO 4** | Be able to use analytical models to guide the design of parallel algorithms. |

|  |
| --- |
| **Practical - I (Code : 4P1)** |
| **CO 1** | **Understand the different types of algorithms and their asymptotic behavior.** |
| **CO 2** | **Understand the different types of data structures and their performance characteristics.** |

|  |
| --- |
| **Project (Code : Project)** |
| **CO 1** | Demonstrate knowledge of the subject content. |
| **CO 2** | Apply skills and knowledge to solve problems. |
| **CO 3** | Work independently and as part of a team. |

|  |
| --- |
| **Seminar (Code : Seminar 1, Seminar 2, Seminar 3, Seminar 4)** |
| **CO 1** | Demonstrate critical thinking and analysis skills. |
| **CO 2** | Communicate effectively in oral and written form. |
| **CO 3** | Engage in self-directed learning. |