DEPARTMENT OF COMPUTER SCIENCE

COURSE OUTCOME

Name of Faculty : Dr. Avinash H.Hedaoo **Program :** B.Sc. **Course** : Computer Science **Semester** : I **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.
Name of Faculty : Mr. Sujay Paldhikar **Program :** B.Sc. **Course** : Computer Science **Semester** : I **Paper** – II : Introduction to Information Technology	
CO1	Students will understand the basic principles of Information Technology like computer-based 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.
Faculty Name : Dr. Avinash H. Hedaoo **Program :** B.Sc. **Course** : Computer Science **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.
Faculty Name : Mr. Sujay Paldhikar **Program :** B.Sc. **Course** : Computer Science **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 self-sufficiency 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.
Faculty Name : Dr. P. V. Nimbalkar **Program :** B.Sc. **Course** : Computer Science **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.
Faculty Name : Dr. P. V. Nimbalkar **Program :** B.Sc. **Course** : Computer Science **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

Faculty Name : Dr. Avinash H. Hedaoo
Program : B.Sc.
Course : Computer Science
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 specifiers 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.

Faculty Name : Mr. Sujay Paldhikar **Programme** : B.Sc. **Course** : Computer Science. **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)
Faculty Name : Dr. Avinash H. Hedaoo **Programme** : B.Sc. **Course** : Computer Science. **Paper – I** : Visual Basic Programming. **Semester-** V	
	Course Outcomes(COs)
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
Faculty Name : Dr. P. V. Nimbalkar **Programme** : B.Sc. **Course** : Computer Science. **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.

Faculty Name : Dr. P. V. Nimbalkar **Programme** : B.Sc. **Course** : Computer Science. **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.
Faculty Name : Dr. P. V. Nimbalkar **Programme** : B.Sc. **Course** : Computer Science. **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