SUBJECT: **COMPUTER SCIENCE (B.Sc.)**	*After completion of the course students will be able to:*
PROGRAMMEOUTCOMES	Computer Science is the scientific and practical approach to computation and its applications. It involves the study of algorithms, data structures, software, hardware, and the underlying principles of computing. The goal is to understand how to solve problems efficiently and effectively using computers. PO1. Scientific Knowledge: This outcome emphasizes the application of fundamental scientific principles, mathematical techniques, and computational methods to solve intricate problems. PO2.Problem Analysis: This involves recognizing and defining complex problems, researching existing solutions, and analyzing them using mathematical and scientific principles. PO3. Design/Development of Solutions: Graduates should be able to design effective solutions and systems that address complex issues PO4.Modern tools usage: Graduates should be skilled in using contemporary tools and techniques for scientific tasks, including prediction and modeling. PO6.The Software Engineer and Society: This outcome highlights the need for graduates to understand and address the broader societal, health, safety, legal, and cultural issues related to their professional work, ensuring responsible practice. PO7.Project management: Demonstrate the scientific and management principles andapply these to one's own work, as a member and leader in a team, to manage projectsand in multidisciplinary environments. PO8. Life-long Learning Life-long learning is a crucial competency that underscores the importance of continually acquiring new knowledge and skills throughout one's career and personal life.
PROGRAMME SPECIFIC OUTCOMES *ForComputer Science (Honors)*	A graduate with a B.Sc.in Computer Science will have the ability to PSO1: Demonstrate core knowledge in the following areas: • Data Structures and Programming Languages • Databases, Software Engineering, and Development • Computer Architecture and Security

	PSO2: Demonstrate problem-solving skills and the application of computer science knowledge to solve real-world problems.
COURSE OUTCOMES:	*Aftercompletion of thecoursethe students will be ableto:*
SEMESTER-I	
CORE-I **Programming Using "C"**	CO1. Grasp fundamentals of C programming, explore various programming constructs, and write C programs utilizing operators and control structures. CO2. Develop C programs utilizing pointers and arrays, and perform pointer arithmetic. CO3. Implement code reusability through functions, manage dynamic memory allocation, and handle command line arguments. CO4. Manage files using different file handling mechanisms, and solve problems employing derived data types.
CORE-II **Digital Logic**	CO1: Define different logic gates, illustrate the realization of Boolean expressions in SOP and POS form, and design these using logic gates. CO2: Design logic circuits such as adders and subtractors. CO3: Design and test combinational circuits. CO4: Design and develop sequential circuits.
GE-I **COMPUTER FUNDAMENTAL**	CO1: Understand the definition and data representation of a computer. CO2: Know the different devices and memory of a computer. CO3: Apply concepts of computer organization and architecture in practical life. CO4: Explain recent emerging technologies and their applications.
SEMESTER-II	
Core Paper III **Programming UsingC++**	CO1: Understand the difference between structure-oriented programming and object-oriented programming. CO2: Apply various object-oriented features such as classes, objects, constructors, and destructors to solve computing problems using C++. CO3: Understand and apply concepts of inheritance and

	operator overloading. CO4: Write programs that perform various operations on files.
Core Paper IV **Data Structure**	CO1: Implement performance analysis of algorithms and various operations on arrays and linked lists. CO2: Implement basic operations of stacks and queues to solve real-world problems. CO3: Implement data representation using trees for various real-life applications. CO4: Implement various sorting algorithms to solve real-world problems.
GE-II **C and DataStructure**	CO1: Implement and formulate algorithms for programs (in C language) and develop programs using the basic elements like control statements. CO2: Implement modular programming approaches and recursion mechanisms to solve complex problems. CO3: Implement programs with pointers and use pre-processors. CO4: Implement the basic operations of stacks and queues and various sorting algorithms to solve real-world problems.
SEMESTER-III	
Core Paper V **OperatingSystem**	CO1: Understand and implement the differences between different types of modern operating systems, virtual machines, their structure, and applications. CO2: Understand the differences between processes and threads, issues of scheduling user-level processes/threads, and the use of locks, semaphores. CO3: Understand and implement concepts of deadlock in operating systems, and how they can be managed/avoided in multiprogramming systems. CO4: Understand and implement the design and management concepts, issues, and challenges of main memory, virtual memory, and file systems.
Core Paper VI **Database System**	CO1: Implement the basics of database management systems. CO2: Implement Structured Query Language (SQL) for database creation and manipulation. CO3: Implement and demonstrate the working of different concepts of DBMS.

	CO4: Implement a database using data definition, data manipulation, and control languages. CO5: Implement and test a project developed for an application, and apply mathematical and formal techniques for solving problems in computer science related to database applications.
Core Paper VII **Discrete Mathematical Structures**	CO1: Apply statements using propositional and predicate logic, prove theorems using mathematical induction, and understand sets, functions, and relations and their properties. CO2: Apply counting principles, permutations, combinations, and the pigeonhole principle to solve counting problems, and solve linear and non-linear recurrence relations using generating functions. CO3: Apply principles and concepts of graph theory to solve real-world problems. CO4: Apply and model DFAs, NFAs, grammars for different languages, minimize DFAs, and apply the pumping lemma to prove a language is not regular.
GE-III **Programming in python**	CO1. Implement Basic Python Syntax and Programming Constructs: Students will be able to implement and understand Python syntax, control structures (such as loops and conditionals), and basic programming constructs like functions and data types. CO2. Develop and Debug Python Programs: Students will be able to develop, test, and debug Python programs using standard libraries and modules, applying best practices in coding and software development. CO3. Apply Object-Oriented Programming (OOP) Concepts: Students will be able to apply OOP principles by creating and utilizing classes and objects in Python, understanding inheritance, polymorphism, and encapsulation. CO4. Utilize Python for Data Manipulation and Analysis: Students will be able to utilize Python libraries such as NumPy, pandas, and Mat plot lib to perform data manipulation, analysis, and visualization. CO5. Build Real-World Applications Using Python: Students will be able to design, implement, and deploy real-world applications using Python, including web

	development with frameworks like Flask or Django, and automation scripts.
SEMESTER-IV	
Core Paper VIII **Java Programming**	CO1: Implement basic concepts of OOP, and introduction to classes and objects through Java Language. CO2: Implement the concepts of constructors, overloading, parameter passing, access control, and inheritance. CO3: Implement the use of packages and interfaces. CO4: Implement exception handling, threads, and access and manipulate databases.
Core Paper IX **Computer Networks**	CO1: Understand various types of signals, transmissions, multiplexing, and networks. CO2: Understand error detection and error correction techniques. CO3: Understand IPv4 and IPv6 and various transport layer protocols. CO4: Understand email and protocols used to transfer data.
Core PaperX **Computer Graphics**	CO1: Apply the background processes involved in computer graphics displays, understanding of algorithms. CO2: Apply mathematics in vectors, create segments, and apply clipping to different shapes. CO3: Apply algorithms used in computer graphics. CO4: Apply methods suitable for 2D and 3D transformations such as translation, rotation, scaling, reflection, and shear. CO5: Apply clipping algorithms for viewing transformation.
GE-IV **Web Technology**	CO1: Develop simple webpages using HTML and Cascading Stylesheets. CO2: Develop web pages using DHTML and Cascading Stylesheets. CO3: Develop dynamic webpages using JavaScript (client-side programming). CO4: Develop interactive web applications using PHP.
SEMESTER-V	
Core Paper XI **Web Technologies**	CO1: Develop simple webpages using HTML and Cascading Stylesheets. CO2: Develop web pages using DHTML and Cascading Stylesheets. CO3: Develop dynamic webpages using JavaScript (client-side programming). CO4: Develop interactive web applications using PHP.
Core Paper XII **Software Engineering**	CO1: Apply the ability to gather and specify requirements of software projects. CO2: Apply the ability to analyze software requirements

	with existing tools. CO3: Apply the understanding of and basic project management practices in real-life projects. CO4: Apply the ability to differentiate different testing methodologies.
DSC-1 **Numerical Techniques**	CO1: Apply knowledge of computer arithmetic and truncation errors in detail. CO2: Apply numerical techniques to find the roots of algebraic equations and check the accuracy of the solutions. CO3: Apply various interpolating methods and several numerical methods to real-life problems. CO4: Apply numerical methods to find numerical integration and numerical solutions of ordinary differential equations
DSC-2 **Unix Shell Programming**	CO1: Learn the basics of UNIX OS, UNIX commands, and the file system. CO2: Learn about the Linux environment. CO3: Learn the fundamentals of shell scripts and shell programming. CO4: Learn to write simple programs using UNIX.
SEMESTER-VI	
CorePaperXIII **Artificial Intelligence**	CO1: Develop an understanding of the basic concepts of AI principles and approaches. CO2: Develop a basic understanding of the building blocks of AI. CO3: Develop the ability to represent knowledge. CO4: Develop an understanding of the basic concepts of Natural Language Processing..
CorePaperXIV **Algorithm Design Techniques**	CO1: Apply sorting algorithms, analyze the efficiency of algorithms using asymptotic notations, and argue the correctness of algorithms using loop invariants. CO2: Apply the concept of hashing, describe and apply the divide-and-conquer paradigm, and derive and solve recurrences describing the performance of divide-and-conquer algorithms. CO3: Apply greedy and dynamic programming algorithms, and solve and analyze several problems using greedy and dynamic programming techniques. CO4: Apply major graph algorithms and analyze their time complexity.
DSEIII: **DataScience**	CO1: Apply knowledge gained from various courses to do innovative work. CO2: Apply knowledge of the complete project lifecycle, project time estimation, and project management. CO3: Apply knowledge of various simulation tools. CO4: Apply skills to work effectively in a team.
DSEIV: **Project Work /Dissertation** **OR**	CO1: Develop innovative work by applying the knowledge gained from various courses undertaken in earlier years.

Data Mining	CO2: Develop an understanding of the complete project lifecycle, project time estimation, and project management. CO3: Develop knowledge of various simulation tools. CO4: Develop the ability to work effectively in a team.