

## DEPARTMENT OF COMPUTER SCIENCE

### B.Sc. in Computer Science (Major-Minor)

#### Programme Specific Outcome

**PO1:** It provides students with a solid grounding in the fundamental principles of computer science, including programming, algorithms, data structures, software engineering, and computer architecture.

**PO2:** Through coursework and practical projects, students develop strong problem-solving abilities essential for addressing complex technological challenges.

**PO3:** Students gain proficiency in various programming languages and paradigms, equipping them to write efficient, maintainable code and develop software applications across different domains.

**PO4:** The course enables students to understand the design, implementation, and management of computer systems, including operating systems, databases, networks, and distributed systems

**PO5:** Depending on the program structure, students may have the chance to specialize in areas such as artificial intelligence, cybersecurity, data science, human-computer interaction, or software engineering.

**PO6:** BSc in Computer Science programs often incorporate practical experiences, such as internships or cooperative education, to prepare students for careers in the tech industry. They may also provide opportunities for collaboration with industry partners on real-world projects.

**PO7:** Given the rapid pace of technological advancement, the curriculum emphasizes adaptability and lifelong learning, ensuring that graduates can continue to evolve and thrive in a dynamic field.

### Course Outcomes of B.Sc. (Computer Science), Major-Minor

Semester	Course Category	Paper Code and Course Name	Outcomes
1 <sup>st</sup>	Major/Minor	<b>COM010104</b>  <b>Computer Fundamentals and Programming</b>	<p><b>This course will enable the students to:</b></p> <p>CO1: Introduction to Computers like Characteristics of computers, Types and generations of computers, Basic operations, Basic components, Interconnecting the units of a computer system, Number system: Representation of numbers and characters in computer, Signed magnitude.</p> <p>CO2: Introduction to computer languages and characteristics, Computer translators, Able to formulate algorithms, Bugs and its types, pseudo codes and</p>

flowcharts take an algorithmic and logical approach to problem solving, Structured programming concepts,

CO3: Understand the basics of C programming and actions, Basic data types in C, application or software program, and instruct them on how to perform, variable, loops, conditional, input, output, subroutines.

CO4: Control structures in C, Decision making with if, if-else, switch statements, Introduction to functions- Function prototypes and arguments, Introduction to recursion. Importance of main () function, return type of main () function.

CO5: Introduction to arrays- Declaration and initialization of arrays, Accessing array elements, Dimension of arrays, Introduction to strings- Declaration and initialization of strings, String input and output in C, String manipulation functions in C

CO6: Introduction to Pointers- Pointer declaration and initialization, Pointers and addresses, Pointers and arrays. Pointers and functions, Introduction to dynamic memory allocation- Allocation and deallocation of memory using malloc(), calloc(), and free() functions.

CO7: Introduction to Structure and Union- Declaration and initialization, Accessing members, Differences between structures and unions

CO8: Introduction to file handling- Opening and closing files, Modes of opening a file, Binary files and text files, Reading and writing files, File pointers, Error handling in file operations, Preprocessor directives, Using preprocessor directives to define constants and macro., Header files

		<p style="text-align: center;"><b>SEC0101303</b></p> <p style="text-align: center;"><b>Basic Programming in C</b></p>	<p><b>This course will enable the students to:</b></p> <p>CO1: Learn the Origin and development of the C programming language, Significance and applications of C in the computing world.</p> <p>CO2: Learn about Basic features and characteristics of C, Advantages and limitations of using C for programming.</p> <p>CO3: Understand the structure of a C program, Components: Header files, main function, variables, and statements.</p> <p>CO4: Learn about Fundamental data types: int, float, char, double, Variable declaration, initialization, and assignment.</p> <p>CO5: Learn how to use printf() and scanf() for standard input and output. Formatting output using format specifiers.</p> <p>CO6: Learn about Addition, subtraction, multiplication, division, modulus. Comparison operators: ==, !=, &lt;, &gt;, &lt;=, &gt;=. Logical operators: &amp;&amp;,   , !. Understanding the assignment (=) operator. Increment (++) and decrement (-) operators.</p> <p>CO7: Understand about if, if-else, nested if-else. switch case statements.</p> <p>CO8: Understand about while loop, do-while loop, for loop. Loop control statements: break, continue.</p> <p>CO9: Learn about calling functions in C. Function prototypes and declarations. Call by value: Passing values to functions Call by reference: Passing addresses to functions using pointers. Creating custom functions to perform specific tasks. Understanding function return types and values.</p> <p>CO10: Understand one-dimensional and multi-dimensional arrays.</p>
--	--	---	---

			<p>Array initialization, accessing elements, and operations.</p>
--	--	--	--

CO11: Learn about strings in C, String functions and operations: strlen(), strcpy(), strcat(), strcmp().

CO12: Understand file handling concepts.

Basic file operations: Reading from and writing to files.

2 <sup>nd</sup>	Major/Minor	<p style="text-align: center;"><b>COM020104</b></p> <p style="text-align: center;"><b>Computer Organization</b></p>	<p><b>This course will enable the students to:</b></p> <p>CO1: Learn about Computer Organization and Architecture along with history. Gain knowledge about Basic functional blocks of a computer:</p> <p>CO2: Learn about number system, and how can they be represented, in terms of fix-point and integers.</p> <p>CO3: Learn about Register and introduction to Register transfer language. Learn about arithmetic micro-operations.</p> <p>CO4: Learn about Instruction codes, Computer registers, General register organization, Register stack, Memory stack, Computer instructions, Data path in a CPU, Operations of a control unit, Hardwired control unit, Micro-programmed control unit.</p> <p>CO5: Able to learn about Memory Organization, Semiconductor memories, Memory cells, Concept of cache memory, Mapping methods, Organization of a cache memory unit.</p> <p>CO6: Learn about Input/output Organization, how to access I/O devices, interrupts and types of interrupts.</p> <p>CO7: Learn about microprocessors, 8085 microprocessor and its operations. Learn about assembly language and its executions.</p>
-----------------	-------------	---	--

		<p style="text-align: center;"><b>CSC-SEC-0206203</b></p> <p style="text-align: center;"><b>HTML</b></p>	<p><b>This course will enable the students to:</b></p> <p>CO1: Learn about HTML, the Head, the Body, Colors, Attributes, Check box, Radio Button, Text, TextArea, Lists, ordered and unordered.</p> <p>CO2: Learn about New Paragraph, Line Break, Blank Space, Preformatted text, Div element Bold text, Important text, Italic text, Emphasized text, Marked text, Small text, Deleted text, Inserted text, Subscript text, Superscript text, HTML quotations, HTML Comments, HTML colors.</p> <p>CO3: Learn about Relative Links, Absolute Links, Link Attributes, Using the ID Attribute to Link within a Document.</p> <p>CO4: Learn how to Put an Image on a Page, Using Images as Links, Putting an Image in the Background.</p> <p>CO5: Learn how to Creating a Table, Table Headers, Captions, Spanning Multiple Columns, Styling Table.</p> <p>CO6: Understand Basic Input and Attributes, Other Kinds of Inputs, Styling forms with CSS, Where To Go From Here</p>
--	--	--	---

3rd	Major/Minor	<p><b>COM030104</b></p> <p><b>Object Oriented Programming using C++</b></p>	<p><b>This course will enable the students to:</b></p> <p>CO1: Learn about Structured programming and Object-Oriented Programming.</p> <p>CO2: Design, Benefits and applications of OOP.</p> <p>CO3: Differentiate between Structured programming and Object-Oriented Programming.</p> <p>CO4: Learn the concept of objects, creating objects.</p> <p>CO5: Learn Nesting of member functions, private member functions, memory allocation for objects.</p> <p>CO6: Imagining real life concepts as objects and derive their properties and functions to operate these objects.</p> <p>CO7: Learn Concept of Inheritance and types of inheritances.</p> <p>CO8: Learn Concept of polymorphism, Compile time and Runtime polymorphism.</p> <p>CO9: Learn Concept of exception handling.</p> <p>CO10: Understand Examples of exceptions and handling exceptions using try, catch and throw statements.</p> <p>CO11: Learn about Abstraction in C++, types of abstraction, abstraction using classes, abstraction in header file, abstraction using access specifiers.</p>
-----	-------------	---	--

4th	Major/Minor	<b>COM040104</b>  <b>Data Structure</b>	<p><b>At the end of the course, students will be able to:</b></p> <p>CO1: Understand and apply the fundamental data structures and algorithms.</p> <p>CO2: Concepts of Data Types, Abstract Data Type, Data Structure, Fundamental and Derived Data Types, Importance of data structures.</p> <p>CO3: Learn Arrays, Array as a data structure, Representation of arrays, Insertion and deletion in arrays. Use of arrays for large number, representation.</p> <p>CO4: Learn linked lists. Singly linked list, doubly linked list, circular linked list. Operations on lists.</p> <p>CO5: Learn Stack and Queue, Representation of stacks and queues using arrays and linked lists.</p> <p>CO6: Learn about Trees, General tree and Binary tree, Basic terminologies, Binary tree traversal methods, Definition and characteristics of threaded binary trees.</p> <p>CO7: Learn about Linear and binary search, Hash Functions, Different Sorting algorithms.</p> <p>CO8: Learn about algorithm, How to analyse a algorithm, Complexity measures of an algorithm(Time and Space), Asymptotic notation as a measure of algorithm complexity, Analysis of sorting algorithms and Searching algorithms in terms of time and space complexity.</p> <p>CO9: Learn to Develop efficient algorithms to solve various computational problems by utilizing data structures and algorithms covered in the course.</p>
-----	-------------	---	---



		<p style="text-align: center;"><b>COM040204</b></p> <p style="text-align: center;"><b>Database Management System</b></p>	<p><b>On successful completion of this course, the student should be able to:</b></p> <p>CO1: Learn database concepts, learn about Database Management System (DBMS), Functions of DBMS, Components of a DBMS, Advantages of DBMS, Disadvantages of Database Systems.</p> <p>CO2: Learn about architecture of DBMS, Schemas, sub-schema and Instances, Data Independence, Three-tier architecture of DBMS, Data models.</p> <p>CO3: Describe different data models used for designing a database, Learn about E-R model, Entity, Attributes, Entity Sets, Domain. Gain Concept of Relationship and Relationship sets, Learn different keys.</p> <p>CO4: Learn about Relational Model and Relational Algebra, Definition of Relation, Data Structure of Relational Database: Relation, Tuples, Attributes Domain, Degree and Cardinality, Operations in Relational Algebra, Different joins.</p> <p>CO5: Learn about Normalization, How to Normalize a database into various normal forms, Benefits of Normalization, Types of Normal Forms.</p> <p>CO6: Learn about Database Languages, Characteristics of SQL, SQL Constraints.</p>
		<p style="text-align: center;"><b>COM040304</b></p> <p style="text-align: center;"><b>Mathematical Foundation of Computer Science</b></p>	<p><b>After successful completion of this course, students will be able to:</b></p> <p>CO1: Learn the concepts of set, relation, and function from Computer Science point of view.</p> <p>CO2: Learn about combinatorics, Basic of counting principles, Mathematical Induction. Pigeonhole principle, generalized Pigeonhole principle and its application.</p> <p>CO3: Learn about graphs, Basic Definition of graph, Representation of graphs in Computers, Algorithms on graph traversals.</p>

CO4: Learn about different Mathematical logics, Connectives, truth tables, Tautologies and Contradictions, Equivalence and Implications, NAND and NOR, Normal forms- CNF, DNF, Converting expressions to CNF and DNF, Theory of inference, Propositional Calculus, Predicate calculus (only introduction), predicates and quantifiers.

**After successful completion of this course, students will be able to:**

CO1: Learn about operating systems, system software, Process Management, Memory Management, File Management,

CO2: Learn about process and threads, process states, new, running, waiting, ready and terminated, Process Control Block, (PCB), information stored in PCB, scheduling queue: job queue, ready queue and device queue, schedulers:

CO3: Learn about process synchronization, Race condition, critical section problem, Peterson's algorithm, Bakery algorithm, synchronization hardware: locking, synchronization software tools.

CO4: Learn about deadlock, Deadlock, operations of a process performs while using a resource: Request. Use and Release, physical and logical resources, Necessary conditions: mutual exclusion, hold & wait, no pre-emption and circular wait, resource allocation graph, deadlock prevention: definition.

CO5: Learn about memory management, Memory hierarchy, base register, limit register, address binding, logical and physical address spaces, memory management unit, relocation register, swapping, contiguous memory allocation: definition, memory protection, fixed partition scheme, variable partition scheme, first-fit, best-fit & worst-fit allocation strategies, non-contiguous memory allocation.

**COM040404**

**Operating System**

--	--	--	--