B.Tech. in Computer Engineering
Semester 3

<table>
<tr><td>I</td><td>Course Code</td><td colspan="4">CS 232001</td></tr>
<tr><td>II</td><td>Course Title</td><td colspan="4">Data Structures</td></tr>
<tr><td>III</td><td>Credit Structure</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td></td><td></td><td>3</td><td>0</td><td>3</td><td>4.5</td></tr>
<tr><td>IV</td><td>Prerequisite</td><td colspan="4">Computer Programming</td></tr>
<tr><td>V</td><td>Learning Outcome</td><td colspan="4">after completing this course, the students will be able to
understand the design of linear and non-linear data structures
implement the linear and non-linear data structures
select appropriate data structure for variety of applications</td></tr>
<tr><td>VI</td><td>Course Content</td><td colspan="4">Introduction to data structures: abstract data types, types of data structures: linear and non-linear data structures;

Linear data structures: arrays, stack, queue, linked list, set, dictionary and hash table; design, implementation and applications;

Non-linear data structures: Hierarchical data structures - tree, binary tree, heap, binary search tree, balanced search trees; design, implementation and applications ;
Graph data structures: design of undirected and directed graphs; graph traversal and search algorithms; implementation and applications

Introduction to external storage structures: hashed indexed files; multi-way trees; b-tree;</td></tr>
<tr><td>VII</td><td>Text/References</td><td colspan="4">Data Structures and Algorithms by Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman
Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein , PHI
Data Structures and Algorithms in Python, Goldwasser, Tamassia and Goodrich, Wiley
Data Structures Through C by Yashavant Kanetkar, BPB publisher
An Introduction to Data Structures with Applications , Jean-Paul Tremblay & Paul G. Sorenson, Tata McGraw Hill</td></tr>
</table>

B.Tech. in Computer Engineering
Semester 3

<table>
<tr><td>I</td><td>Course Code</td><td colspan="4">CS 232002</td></tr>
<tr><td>II</td><td>Course Title</td><td colspan="4">Object Oriented Programming Concepts</td></tr>
<tr><td>III</td><td>Credit Structure</td><td>L</td><td>T</td><td>P</td><td>C</td></tr>
<tr><td></td><td></td><td>3</td><td>0</td><td>3</td><td>4.5</td></tr>
<tr><td>IV</td><td>Prerequisite</td><td colspan="4">Computer Programming</td></tr>
<tr><td>V</td><td>Learning Outcome</td><td colspan="4">after completing this course, the students will be able to
• understand the fundamental principles and concepts of object-oriented programming, including encapsulation, inheritance, and polymorphism.
• Students will learn how to design and implement classes, which are the building blocks of object-oriented programming</td></tr>
<tr><td>VI</td><td>Course Content</td><td colspan="4">The topics covered in this course:
• Introduction to Object Oriented Programming
• Classes and Methods ○Polymorphism
• Inheritance
• Standard Library of C++

1. Beginning with OOP Language: Review of Tokens, Expressions, Operators & Control Structures. Scope Resolution Operator, Member Dereferencing Operator, Reference Variables.
2. Review of Functions, Function Overloading, Inline Functions, Default Arguments.
3. Classes & Objects: Specifying a Class, Defining Member Functions, creating Class Objects, Accessing Class Members. Access Specifies – Public, Private, and Protected Classes, Its Members, Objects and Memory Allocation

4. Static Members, the Const Keyword and Classes, the Static Objects. Friend Function & its Usage Empty Classes, Nested Classes, Local Classes.
5. Constructors & Destructors: Need for Constructors and Destructors, Copy Constructors, Dynamic Constructors, Destructors, Constructors and Destructors with Static Members.
6. Operator Overloading & Type Conversion: Defining Operator Overloading, Rules for Overloading Operators, Overloading of Unary Operators and various Binary Operators with Friend Functions and Member Functions. Type Conversion – Basic Type to Class Type, Class Type to Basic Type, Class Type to another Class Type.
7. Inheritance: Introduction, Defining Derived Classes, Forms of Inheritance, Ambiguity in Multiple and Multipath Inheritance, Virtual Base Class, Overriding Member Functions, Order of Execution of Constructors and Destructors Virtual Functions & Polymorphism: Virtual Functions, Pure Virtual Functions, Abstract Classes, Introduction to Polymorphism
8. Pointers & Dynamic Memory Management: Understanding Pointers, Accessing Address of a Variable, Declaring & Initializing Pointers, Pointer to a Pointer, Pointer to a Function, Dynamic Memory Management – New and Delete Operators, this Pointer.
9. Console I/O: Concept of Streams, Hierarchy of Console Stream Classes,</td></tr>
</table>

		Unformatted I/O Operations, Managing Output with Manipulators. 10. Working with Files: Opening, Reading, Writing, Appending, Processing & Closing different Type of Files, Command Line Arguments 11. Standard Template Library (STL) of C++
VII	Text/References	Text Books: 6. Object Oriented Programming with C++ by E. Balagurusamy, McGraw-Hill Education (India) 7. ANSI and Turbo C++ by Ashoke N. Kamthane, Pearson Education Reference Books: 8. Big C++ - Wiley India 2. C++: The Complete Reference- Schildt, McGraw-Hill Education (India) 9. C++ and Object Oriented Programming – Jana, PHI Learning. 10. Object Oriented Programming with C++ - Rajiv Sahay, Oxford 11. Mastering C++ - Venugopal, McGraw-Hill Education (India) 12. "Accelerated C++: Practical Programming by Example" by Andrew Koenig and Barbara E. Moo:

B.Tech. in Computer Engineering
Semester 3

I	Course Code	CS 232003			
II	Title of the course	Digital Logic Design			
III	Credit Structure	L	T	P	C
		3	0	3	4.5
IV	Prerequisite	Nil			
V	Learning Outcome	after completing this course, the students will be able to • understand the concepts Boolean algebra and digital logic design • use various techniques to reduce logical expressions using Boolean algebra and k-map. • design and implement combinational and sequential circuits, memory and programmable logic for carrying out specified tasks			
VI	Course Content	**Digital systems and binary numbers:** binary, octal, hexadecimal numbers; base conversion and complements of numbers; signed and unsigned numbers; binary codes; binary storage and registers; **Boolean algebra and logic gates:** postulates, theorems and properties of Boolean algebra; Boolean functions, canonical and standard forms; logic gates and integrated circuits; minimization of Boolean functions using algebraic, Karnaugh map and Quine – McClausky methods; product of sum simplification; NAND and NOR implementation, Exclusive-OR function; Introduction to Hardware Description Language; **Combinational Circuits:** design procedure; design of adder, subtractor, multiplier, comparator, decoders and encoders, multiplexers, analysis of combinational circuits; HDL models for combinational circuits; **Synchronous sequential logic:** sequential circuits; storage elements latches and flip-flops; SR, JK, D, T flip-flops; clocked sequential circuits; master-slave flip-flop, edge-level triggering considerations; HDL models for sequential circuits and design procedure; registers and counters; HDL models for registers and counters; **Memory and programmable logic:** random access memory, memory decoding, error detection and correction, read only memory, programmable logic arrays and sequential programmable devices; **Design at register transfer level:** RTL notation, algorithmic state machines; sequential binary multiplier, control logic, design of multiplexers, HDL description of design examples; race condition free design; introduction to System Verilog;			
VII	Text/Reference books:	1. Digital Design – with an introduction to the Verilog HDL, VHDL, and SystemVerilog by M. Morris Mano and Michael D Cilette,			

		Pearson publisher 2. Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic , Mc Graw Hill 3. Digital Systems: Principles and Applications by Tocci, R. J., Widmer, N. S., & Moss, G. L., Pearson publisher 4. Digital Fundamentals by Floyd, T. L. Pearson Education India. 5. Digital System Design using VHDL Roth, C. H. Jr., Cengage Learning/ PWS publishing

I	Course Code	**MA 232001**
II	Course Title	**Discrete Mathematics**
III	Credit Structure	L T P C 3 1 0 4
IV	Prerequisites (If any)	
V	Learning Outcome	after completing this course, the students will be able to • Basics of Discrete Mathematics which comprises the essentials for a computer science student to go ahead and study any other topics in the subject. The emphasis will be on problem-solving as well as proofs.
VI	Course Content	• Basic logic: Propositional logic: logical connectives; truth tables; normal forms (conjunctive and disjunctive); validity; predicate logic; limitations of predicate logic, universal and existential quantification; modus ponens and modus tollens. Proof techniques: Notions of implication, converse, inverse, contrapositive, negation, and contradiction; the structure of formal proofs; direct proofs; proof by counterexample; proof by contraposition; proof by contradiction; mathematical induction; strong induction; recursive mathematical definitions; well orderings. • Basics of counting: Counting arguments; pigeonhole principle; permutations and combinations; inclusion-exclusion, recurrence relations, generating functions. • Fundamental structures; Functions (surjections, injections, inverses, composition); relations (reflexivity, symmetry, transitivity, equivalence relations); sets (Venn diagrams, complements, Cartesian products, power sets); pigeonhole principle; cardinality and countability. • Graph Theory, Connectivity Euler and Hamiltonian paths, shortes path.
VIII	Text/References	1. Discrete Mathematics and Its applications, Kenneth Rosen,