SREEPATHY INSTITUTE OF MANAGEMENT AND TECHNOLOGY

COURSE HANDBOOK

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SEMESTER III

MAT203 DISCRETE MATHEMATICAL STRUCTURES

COURSE INFORMATION SHEET:

Program: **Computer Science Engineering**	Degree: **B-Tech**
Course**:** Discrete Mathematical Structures	Course code**: MAT 203**
L-T-P**:3-1-0**	Credit**:4**

SYLLABUS:

MODULE	CONTENT	HOURS	UNIVERSITY % MARKS
I	Mathematical logic - Basic connectives and truth table, Statements, Logical Connectives, Tautology, Contradiction. Logical Equivalence - The Laws of Logic, The Principle of duality, Substitution Rules. The implication - The Contrapositive, The Converse,The Inverse. Logical Implication - Rules of Inference. The use of Quanttifiers S- Open Statement, Quantifier. Logically Equivalent – Contrapositive, Converse , Inverse , Logical equivalences and implications for quantified statement, Implications , Negation	9	20
II	The Rule of Sum – Extension of Sum Rule . The Rule of Product - Extension of Product Rule . Permutations. Combinations. The Binomial Theorem (without proof). Combination with Repetition. The Pigeon hole Principle. The Principle of Inclusion and Exclusion Theorem (Without Proof) - Generalization of the Principle. Derangements.	9	20
III	Cartesian Product - Binary Relation. Function – domain, range-one to one function, Imagerestriction. Properties of Relations- Reachability Relations, Reflexive Relations, Symmetric Relations, Transitive relations, Anti-symmetric Relations, Partial Order relations, Equivalence Relations, Irreflexive relations. Partially ordered Set – Hasse Diagram, Maximal-Minimal Element, Least upper bound (lub), Greatest Lower bound(glb) (Topological sorting Algorithm- excluded). Equivalence Relations and Partitions- Equivalence Class. Lattice - Dual Lattice, Sub lattice, Properties of glb and lub , Properties of Lattice, Special Lattice, Complete Lattice, Bounded Lattice, Completed Lattice , Distributive Lattice.	9	20
IV	Generating Function - Definition and Examples , Calculation techniques, Exponential generating function. First order linear recurrence relations with constant coefficients – homogeneous, non-homogeneous Solution. Second order linear recurrence relations with constant coefficients, homogeneous, non-homogeneous Solution.	9	20
V	Algebraic system-properties- Homomorphism and Isomorphism. Semi group and monoid – cyclic monoid,	9	20

	sub semi group and sub monoid, Homomorphism and Isomorphism of Semi group and monoids. Group-Elementary properties, subgroup, symmetric group on three symbols ,The direct product of two groups, Group Homomorphism, Isomorphism of groups, Cyclic group. Right cosets - Left cosets. Lagrange's Theorem		

TEXT BOOKS:

1	Discrete and Combinatorial Mathematics (An Applied Introduction), Ralph P Grimaldi, B V Ramana , 5th Edition, Pearson

REFERENCES:

1	Kenneth H. Rosen, Discrete Mathematics and Its Applications with Combinatorics and Graph Theory, Seventh Edition, MGH, 2011
2	Trembly J.P and Manohar R, "Discrete Mathematical Structures with Applications to Computer Science", Tata Mc Graw Hill Pub. Co. Ltd., New Delhi, 2003.
3	Bernard Kolman, Robert C. Busby, Sharan Cutler Ross, "Discrete Mathematical Structures", Pearson Education Pvt Ltd., New Delhi, 2003
4	Kenneth H .Rosen, "Discrete Mathematics and its Applications", 5/e, Tata Mc Graw Hill Pub. Co. Ltd, New Delhi 2003
5	Richard Johnsonbaugh, "Discrete Mathematics", 5/e, Pearson Education Asia, NewDelhi, 2002.
6	Joe L Mott, Abraham Kandel, Theodore P Baker,"Discrete Mathematics for Computer Scientists and Mathematicians", 2/e, Prentice-Hall India, 2009.

PREREQUISITE:

A sound background in higher secondary school Mathematics

COURSE OBJECTIVES:

1	Familiarizing concepts in discrete mathematics that is essential for computing
2	To provide knowledge about strategies of reasoning and proofs in mathematical logic.
3	To provide a thorough understanding for creative and analytical thinking.

COURSE OUTCOMES:

After successful completion of the course, the students should be able to:

CO's	DESCRIPTION
1	Apply truth tables, deductive reasoning and propositional logic to verify the validity of predicates in Propositional and Quantified Propositional Logic
2	Use elementary counting techniques to solve counting problems.
3	Identify various types of binary relations & their Domain specific application for Computer Science.
4	Demonstrate Lattices and Partially Ordered Sets for applications in Computer Science
5	Solve Linear Recurrence Relations upto Second order and Explain Generating Functions.
6	Differentiate different abstract algebraic systems.

CO-PO-PSO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2	PSO3
CO1	3	2	1	-	–	–	–	–	–	–	–	–		3	-	-
CO2	3	3	1	-	–	–	–	–	–	–	–	–		3	-	-
CO3	3	-	-	–	–	–	–	–	–	–	–	3		-	-	-
CO4	3	2	-	–	–	–	–	–	–	–	–	–		3	-	-
CO5	3	2	-	–	–	–	–	–	–	–	–	–		-	-	-
CO6	3	2	2	–	–	–	–	–	–	–	–	–		3	-	-

CO-PO MAPPING JUSTIFICATION:

CO's	PO's	JUSTIFICATION
CO1	**PO1**	The concepts of discrete structures can be used to solve various complex engineering problems.
	PO2	The knowledge about the discrete computational structures will help them to reach conclusions about the complexity and methodologies for solving real life problems
	PO3	Discrete structures can aid in the representation of various real life problems
CO2	**PO1**	The validity of facts can be verified using predicate and propositional logic
	PO2	The real life events can be represented and verified using Mathematical

		logic
	PO3	Reasoning is made possible for engineering problems
CO3	**PO1**	The reasoning and inferences made by them can be substantiated by the various proof techniques
	PO12	The proof techniques can be used to verify the complex engineering solutions
CO4	**PO1**	Algebraic structures can be used to visualize the complex engineering problems involving sets of data
	PO2	The similarity and characteristics of data can be analyzed using algebraic principles
CO5	**PO1**	The arrangement and combinations of data to be taken for different problems can be identified
	PO2	Counting techniques can be used to reach conclusions in the problems involving huge data
CO6	**PO1**	It can be used to compare and contrast the complexity of algorithms that were developed
	PO2	It helps to analyze the complexity and choose the best method for the particular problem
	PO3	All algorithms can be compared using a single measure to identify the amount of computations involved in them so that the optimal one can be identified

CO-PSO MAPPING JUSTIFICATION:

CO's	PSO's	JUSTIFICATION
CO1	**PSO1**	To Apply the Engineering knowledge in mathematical logic related to computer science
CO2	**PSO1**	Explore the technical knowledge for identifying counting problem scenarios
CO4	**PSO1**	To Apply the Engineering knowledge to identify, Analyze, Design mathematical structures
CO6	**PSO1**	Identifying & use the knowledge in Computer Science domain to differentiate the specified structures

CST 201 DATA STRUCTURES

COURSE INFORMATION SHEET:

Program: **Computer Science & Engineering**	Degree :**B-Tech**
Course**: Data structures**	Course code**: CST 201**
L-T-P**: 3-1-0**	Credit**:4**

SYLLABUS:

MODULE	CONTENT	HOURS	UNIVERSITY % MARKS
I	**Basic Concepts of Data Structures** System Life Cycle, Algorithms, Performance Analysis, Space Complexity, Time Complexity, Asymptotic Notation, Complexity Calculation of Simple Algorithms	5	20
II	**Arrays and Searching** Polynomial representation using Arrays, Sparse matrix, Stacks, Queues-Circular Queues, Priority Queues, Double Ended Queues, Evaluation of Expressions , Linear Search and Binary Search	10	20
III	**Linked List and Memory Management** Self Referential Structures, Dynamic Memory Allocation, Singly Linked List-Operations on Linked List. Doubly Linked List, Circular Linked List, Stacks and Queues using Linked List, Polynomial representation using Linked List Memory allocation and de-allocation-First-fit, Best-fit and Worst-fit allocation schemes	12	20
IV	**Trees and Graphs** Trees, Binary Trees-Tree Operations, Binary Tree Representation, Tree Traversals, Binary Search Trees- Binary Search Tree Operations Graphs, Representation of Graphs, Depth First Search and Breadth First Search on Graphs, Applications of Graphs	8	20
V	**Sorting and Hashing** Sorting Techniques – Selection Sort, Insertion Sort, Quick Sort, Merge Sort and Heap Sort Hashing- Hashing Techniques, Collision Resolution, Overflow handling, Hashing functions – Mid square, Division, Folding, Digit Analysis	10	20

TEXT BOOKS:

1.	Ellis Horowitz, Sartaj Sahni and Susan Anderson-Freed, Universities Press, Fundamentals of Data Structures in C

REFERENCES:

1.	Samanta D., Classic Data Structures, Prentice Hall India.
2.	Hugges J. K. and J. I. Michtm, A Structured Approach to Programming, PHI.
3.	Aho A. V., J. E. Hopcroft and J. D. Ullman, Data Structures and Algorithms, Pearson Publication.
4.	Tremblay J. P. and P. G. Sorenson, Introduction to Data Structures with Applications, Tata McGraw Hill.
5.	Peter Brass, Advanced Data Structures, Cambridge University Press.
6.	Lipschuts S., Theory and Problems of Data Structures, Schaum's Series.
7.	Wirth N., Algorithms + Data Structures = Programs,
8.	Martin Barrett, Clifford Wagner, C And Unix: Tools For Software Design, John Wiley.

PREREQUISITE: Topics covered under the course Programming in C (EST 102)

COURSE OBJECTIVES:

1.	To impart a thorough understanding of linear data structures such as stacks, queues and their applications.
2.	To impart a thorough understanding of non-linear data structures such as trees, graphs and their applications.
3.	To impart familiarity with various sorting, searching and hashing techniques and their performance comparison.
4.	To impart a basic understanding of memory management.

COURSE OUTCOMES:

After successful completion of the course, the students should be able to:

CO's	DESCRIPTION
1.	Design an algorithm for a computational task and calculate the time/space complexities of that algorithm (**Cognitive Knowledge Level: Apply)**
2.	Identify the suitable data structure (array or linked list) to represent a data item required to be processed to solve a given computational problem and write an algorithm to find the solution of the computational problem **(Cognitive Knowledge Level: Apply**)
3.	Write an algorithm to find the solution of a computational problem by selecting an appropriate data structure (binary tree/graph) to represent a data item to be processed **(Cognitive Knowledge Level: Apply**)
4.	Store a given dataset using an appropriate Hash Function to enable efficient access of data in the

	given set (**Cognitive Knowledge Level: Apply**)
5.	Select appropriate sorting algorithms to be used in specific circumstances (Cognitive Knowledge Level: **Analyze**)
6.	Design and implement Data Structures for solving real world problems efficiently (**Cognitive Knowledge Level: Apply**)

CO-PO-PSO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2	PSO3
CO 1	3	2	2	1		1						1		2	1	1
CO 2	3	2	2	2		1						1		3	-	-
CO 3	3	3	3	2		1						1		3	1	-
CO 4	3	3	3	2		1						1		2	-	1
CO 5	3	2	2	1		1						1		1	-	-
CO 6	3	3	3	2		1						1		-	-	-

CO-PO MAPPING JUSTIFICATION:

CO s	PO s	JUSTIFICATION
CO1	**PO1**	Apply the knowledge of mathematics in problems like time complexity
	PO2	Identify less complex algorithm.
	PO3	Design the solution by applying best data structure and less complex algorithm.
	PO4	The knowledge of asymptotic notations helps in analysis of performance of solutions to complex problems
	PO6	The knowledge in algorithms help in designing solutions of the societal problems.
	PO12	Recognize the need for algorithm complexity and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change
CO2	**PO1**	Apply the knowledge of mathematics and engineering fundamentals to the solution of problems related to arrays & linked lists.
	PO2	Identifying and analyzing the use of arrays, linked lists in stacks, queues etc
	PO3	The knowledge of arrays, linked lists can be applied to solve complex engineering problems.

	PO4	.Knowledge in algorithms complexity leads to the development of better solutions
	PO6	Various algorithms can be developed for societal needs
	PO12	Recognize the need for linear data structures and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change
CO3	**PO1**	The knowledge of non linear data structures like trees and graphs can be applied to solve complex engineering problems.
	PO2	To analyse different non linear data structures like trees and graphs
	PO3	The knowledge of non linear data structures like trees and graphs can be applied to solve complex engineering problems.
	PO4	The knowledge of non linear data structures helps in representation, analysis and interpretation of data to provide valid conclusions.
	PO6	Algorithms can be developed like that of in google maps, social networking sites that apply the properties of graph and for compression algorithms in router tables etc which uses trees.
	PO12	Recognize the need for non- linear data structures and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change
CO4	**PO1**	The knowledge of various hashing techniques can be applied in designing solutions to complex engineering problems.
	PO2	To analyse different hashing techniques or hash functions to arrive at conclusions
	PO3	The knowledge of hashing can be applied to design solutions to complex engineering problems.
	PO4	The knowledge helps in representation, analysis and interpretation of data to provide valid conclusions.
	PO6	Algorithms can be developed for modern day cryptography hash functions etc
	PO12	Recognize the need hashing techniques and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change
CO5	**PO1**	This basic knowledge of sorting and searching can be used in solutions to complex engineering problems.
	PO2	Identify and analyze engineering problems to arrive at substantiated conclusions regarding sorting and searching
	PO3	This basic knowledge of sorting and searching can be used in designing solutions to complex engineering problems.
	PO4	This concept is fundamental in conducting investigations and interpretations of data.
	PO6	Various algorithms can be developed for societal needs
	PO12	Recognize the need for solving real world problems and have the preparation and ability to engage in independent and lifelong learning in

		the broadest context of technological change

CO-PSO MAPPING JUSTIFICATION:

CO s	PSO s	JUSTIFICATION
CO1	PSO1	This knowledge helps to design good and efficient algorithms and thereby can be used in research and other innovative ideas.
	PSO2	The knowledge in data structure will help to design various less complex networking algorithms
	PSO3	These concepts are fundamental to CS and can be used in research and other innovative ideas.
CO2	PSO2	The knowledge of arrays, linked lists, stacks and queues can be applied to design solutions to complex engineering problems in using suitable data structures
CO3	PSO1	The basic information in non linear data structures like trees and graphs will help in choosing the best data structures and algorithms
	PSO2	The basic information in non linear data structures like trees and graphs will help in building practical proficiency
CO4	PSO1	This basic knowledge of sorting and searching will help in choosing the best data structures and algorithms
	PSO3	The knowledge about sorting and searching like trees and graphs will help in building practical proficiency
CO5	PSO2	This basic knowledge of hashing will help in choosing the best data structures and algorithms

CST 203 LOGIC SYSTEM DESIGN

COURSE INFORMATION SHEET:

Program: **Computer Science & Engineering**	Degree : **B-Tech**
Course**: Logic System Design**	Course code**: CST 203**
L-T-P**:3-1-0**	Credit**:4**

SYLLABUS:

MODULE	CONTENT	HOURS	UNIVERSITY % MARKS
I	Number systems, Operations & Codes Decimal, Binary, Octal and Hexadecimal Number Systems- Number Base Conversions. Addition, Subtraction, Multiplication and Division of binary numbers. Representation of negative numbers- Complements, Subtraction with complements. Addition and subtraction of BCD, Octal and Hexadecimal numbers. Binary codes- Decimal codes, Error detection codes, Reflected code, Character coding schemes – ASCII, EBCDIC.	7	20
II	Boolean Algebra Postulates of Boolean Algebra. Basic theorems and Properties of Boolean Algebra. Boolean Functions - Canonical and Standard forms. Simplification of Boolean Functions- Using Karnaugh-Map Method (upto five variables), Don't care conditions, Product of sums Total Marks CIE Marks ESE Marks ESE Duration 150 50 100 3 simplification, Tabulation Method. Digital Logic Gates- Implementation of Boolean functions COMPUTER SCIENCE AND ENGINEERING using basic and universal gates.	9	20
III	Combinational Logic Circuits Design Procedure & Implementation of combinational logic circuits- Binary adders and subtractors, Binary Parallel adder, Carry look ahead adder, BCD adder, Code converter, Magnitude comparator, Decoder, Demultiplexer, Encoder, Multiplexer, Parity generator/ Checker.	9	20
IV	Sequential logic circuits: Flip-flops- SR, JK, T and D. Triggering of flip-flops- Master slave flip- flops, Edge-triggered flip- flops. Excitation table and characteristic equation. Registers- register with parallel load. Counter design: Asynchronous counters- Binary and BCD counters, timing sequences and state diagrams. Synchronous counters- Binary Up- down counter, BCD counter.	9	20
V	Shift registers Shift registers – Serial In Serial Out, Serial In Parallel Out, Bidirectional Shift Register with Parallel load. Ring counter. Johnson counter- timing sequences and state diagrams. Arithmetic algorithms Algorithms for addition and subtraction of binary numbers in signed	11	20

	magnitude and 2's complement representations. Algorithm for addition and subtraction of BCD numbers. Representation of floating point numbers, Algorithm for addition and subtraction of floating point numbers. Programmable Logic devices ROM. Programmable Logic Array(PLA)- Implementation of simple circuits using PLA.		

TEXT BOOKS:

1	M. Morris Mano, Digital Logic & Computer Design, 4/e, Pearson Education, 2013
2	Thomas L Floyd, Digital Fundamentals, 10/e, Pearson Education, 2009.
3	M. Morris Mano, Computer System Architecture, 3/e, Pearson Education, 2007.

REFERENCES:

1	M. Morris Mano, Michael D Ciletti , Digital Design With An Introduction to the Verilog HDL, 5/e, Pearson Education, 2013
2	Donald D Givone, Digital Principles and Design, Tata McGraw Hill, 2003

PREREQUISITE:

NIL

COURSE OUTCOMES:

After successful completion of the course, the students should be able to:

CO's	DESCRIPTION
1	Illustrate decimal, binary, octal, hexadecimal and BCD number systems, perform conversions among them and do the operations - complementation, addition, subtraction, multiplication and division on binary numbers (Cognitive Knowledge level: Understand)
2	Simplify a given Boolean Function and design a combinational circuit to implement the simplified function using Digital Logic Gates (Cognitive Knowledge level: Apply)
3	Design combinational circuits - Adders, Code Convertors, Decoders, Magnitude Comparators, Parity Generator/Checker and design the Programmable Logic Devices - ROM and PLA. (Cognitive Knowledge level: Apply)
4	Design sequential circuits - Registers, Counters and Shift Registers. (Cognitive Knowledge level: Apply)
5	Use algorithms to perform addition and subtraction on binary, BCD and floating point numbers (Cognitive Knowledge level: Understand)

CO-PO-PSO MAPPING:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12		PSO1	PSO2	PSO3
CO1	**3**	**2**													**3**	
CO2	**2**	**2**	**3**	**1**		**1**								**1**	**3**	
CO3	**2**	**2**	**3**	**1**		**1**								**1**		
CO4	**2**	**2**	**3**	**1**		**1**								**1**		
CO5	**2**	**2**	**3**												**2**	

CO-PO MAPPING JUSTIFICATION:

CO1	**PO1**	**3**	Understand the different number systems by applying the knowledge of mathematics and perform various number conversions required for digital systems
	PO2	**2**	Analyse various operations to be done on digital system by understanding the engineering problems
CO2	**PO1**	**2**	Apply the knowledge of mathematic and engineering fundamentals to perform the simplification of Boolean expressions
	PO2	**2**	Formulate various Boolean expressions in order to design a combinational circuits using logic gates
	PO3	**3**	Design various combinational circuits using gates in order to implement digital system
	PO4	**1**	Use appropriate knowledge from the design to derive valid conclusions for Boolean simplification
	PO6	**1**	Apply reasoning based on contextual knowledge to use the design to solve the solve the safety and other issues of society, keeping in mind the ethical principles of engineering
CO3	**PO1**	**2**	Apply the engineering specializations to design combinational circuits
	PO2	**2**	Apply engineering knowledge to design appropriate combinational circuits
	PO3	**3**	Design various combinational circuits and PLA devices using the design principles of engineering
	PO4	**1**	Use appropriate research based knowledge to implement the design of various combinational circuits and PLA devices
	PO6	**1**	Apply reasoning based contextual knowledge to use the various combinational designs and PLA devices to solve the safety and societal issues based on the ethical principles of engineering
CO4	**PO1**	**2**	Apply the knowledge of science and engineering to design various sequential circuits like registers flip flops and counters

	PO2	**2**	Formulate various designs associated with sequential circuits by applying engineering knowledge
	PO3	**3**	Design various sequential circuits using the design principles of engineering
	PO4	**1**	Use appropriate research based knowledge to implement the design of various sequential circuits
	PO6	**1**	Apply reasoning based contextual knowledge to use the various sequential designs and devices to solve the safety and societal issues based on the ethical principles of engineering
CO5	**PO1**	**2**	Understand the concepts of algorithms by applying the mathematical knowledge and engineering principles
	PO2	**2**	Formulate various algorithms to perform various operations on binary and BCD numbers by applying mathematical and engineering knowledge
	PO3	**3**	Formulate various algorithms by applying engineering knowledge

CO-PSO MAPPING JUSTIFICATION:

CO1	**PSO2**	**3**	Graduates will be able to understand the concepts of number systems using the mathematical and engineering knowledge
CO2	**PSO1**	**1**	Graduates attain the ability to design combinational circuits following professional skills
	PSO2	**3**	Graduates will be able to understand the design principles of digital system
CO3	**PSO1**	**1**	Graduates attain the ability to design combinational circuits following the professional ethics
CO4	**PSO1**	**1**	Graduates will be able to design sequential circuits following the professional ethics
CO5	**PSO2**	**2**	Graduates will be able to understand the concepts of algorithms

CST 205 OBJECT ORIENTED DESIGN USING JAVA

COURSE INFORMATION SHEET:

Program: **Computer Science & Engineering**	Degree :**B-Tech**
Course**: Object Oriented Design using JAVA**	Course code**: CST 205**
L-T-P**:3-1-0**	Credit**:4**

SYLLABUS:

MODULE	CONTENT	HOURS	UNIVERSITY % MARKS
I	Introduction-Approaches to Software Design - Functional Oriented Design, Object Oriented Design, Case Study of Automated Fire Alarm System. Object Modeling Using Unified Modeling Language (UML) – Basic Object Oriented concepts, UML diagrams, Use case model, Class diagram, Interaction diagram, Activity diagram, State chart diagram. Introduction to Java - Java programming Environment and Runtime Environment, Development Platforms -Standard, Enterprise. Java Virtual Machine (JVM), Java compiler, Bytecode, Java applet, Java Buzzwords, Java program structure, Comments, Garbage Collection, Lexical Issues.	8	15
II	Primitive Data types - Integers, Floating Point Types, Characters, Boolean. Literals, Type Conversion and Casting, Variables, Arrays, Strings, Vector class. Operators - Arithmetic Operators, Bitwise Operators, Relational Operators, Boolean Logical Operators, Assignment Operator, Conditional (Ternary) Operator, Operator Precedence. Control Statements - Selection Statements, Iteration Statements and Jump Statements. Object Oriented Programming in Java - Class Fundamentals, Declaring Objects, Object Reference, Introduction to Methods, Constructors, this Keyword, Method Overloading, Using Objects as Parameters, Returning Objects, Recursion, Access Control, Static Members, Final Variables, Inner Classes, Command Line Arguments, Variable Length Arguments. Inheritance - Super Class, Sub Class, The Keyword	11	15

	super, protected Members, Calling Order of Constructors, Method Overriding, the Object class, Abstract Classes and Methods, using final with Inheritance.		
III	Packages and Interfaces - Defining Package, CLASSPATH, Access Protection, Importing Packages, Interfaces. Exception Handling - Checked Exceptions, Unchecked Exceptions, try Block and catch Clause, Multiple catch Clauses, Nested try Statements, throw, throws and finally. Input/Output - I/O Basics, Reading Console Input, Writing Console Output, PrintWriter Class, Object Streams and Serialization, Working with Files.	8	15
IV	Advanced features of Java: Java Library - String Handling – String Constructors, String Length, Special String Operations - Character Extraction, String Comparison, Searching Strings, Modifying Strings, using valueOf(), Comparison of StringBuffer and String. Collections framework - Collections overview, Collections Interfaces- Collection Interface, List Interface. Collections Class – ArrayList class. Accessing a Collection via an Iterator. Event handling - Event Handling Mechanisms, Delegation Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Using the Delegation Model. Multithreaded Programming - The Java Thread Model, The Main Thread, Creating Thread, Creating Multiple Threads, Synchronization, Suspending, Resuming and Stopping Threads.	10	15
V	Graphical User Interface and Database support of Java: Swings fundamentals - Swing Key Features, Model View Controller (MVC), Swing Controls, Components and Containers, Swing Packages, Event Handling in Swings, Swing Layout Managers, Exploring Swings –JFrame, JLabel, The Swing Buttons, JTextField. Java DataBase Connectivity (JDBC) - JDBC overview, Creating and Executing Queries – create table, delete, insert, select.	8	20

TEXT BOOKS:

1	Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.

2	Rajib Mall, Fundamentals of Software Engineering, 4th edition, PHI, 2014.
3	Paul Deitel, Harvey Deitel, Java How to Program, Early Objects 11th Edition, Pearson, 2018.

REFERENCES:

1	Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.
2	Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008.
3	Flanagan D., Java in A Nutshell, 5/e, O'Reilly, 2005.
4	Barclay K., J. Savage, Object Oriented Design with UML and Java, Elsevier, 2004.

PREREQUISITE:

Topics covered under the course PROGRAMMING IN C (EST 102)

COURSE OBJECTIVES:

1	Write Java programs using the object oriented concepts - classes, objects, constructors, data hiding, inheritance and polymorphism (Cognitive Knowledge Level: Apply)
2	Utilise data types, operators, control statements, built in packages & interfaces, Input/ Output Streams and Files in Java to develop programs (Cognitive Knowledge Level: Apply)
3	Illustrate how robust programs can be written in Java using exception handling mechanism (Cognitive Knowledge Level: Understand)
4	Write application programs in Java using multithreading and database connectivity (Cognitive Knowledge Level: Apply)
5	Write Graphical User Interface based application programs by utilising event handling features and Swing in Java (Cognitive Knowledge Level: Apply)

COURSE OUTCOMES:

After successful completion of the course, the students should be able to:

CO's	DESCRIPTION
1	Enables to develop UML diagrams and then implement it as a Java program in accordance with your UML design..
2	Enable to do a Java program to evaluate a postfix expression containing two operands and a single operator using stack. Stack should be implemented as a separate entity so as to reflect OOP concepts.
3	Implements a program to demonstrate the start, run, sleep and join methods in Thread class.
4	Write a GUI based program with separate buttons to add, delete and display

	student details i.e. name, student ID, current semester and branch of study based on student ID.
5	Using Swing creates a JFrame with a JLabel and two JButtons. Set the texts of JButtons as “Yes” and “No” respectively. Set the JLabel’s text to the text of the button currently being pressed. Initially the JLabel’s text is blank.

CO-PO-PSO MAPPING:

CO-PO MAPPING JUSTIFICATION:

	CO’S	PO’S		Justification
CS306	**CO1**	**PO1**	**1**	Apply the knowledge acquired to classify the UML based on its function
		PO2	**2**	Understanding the OOP functions helps the students to identify and formulate the problems based on the concepts
		PO4	**2**	Understanding the functions and understanding the synchronization factors, helps in analyzing and interpreting threads.
	CO2	**PO2**	**1**	Apply the basic knowledge of packages and its working process to develop different packages
		PO4	**2**	Understanding the various database access techniques helps in analyzing and interpreting the oops concepts
	CO3	**PO3**	**2**	Studies about the various layout helps the students to fix up the labels and buttons in different layout
		PO4	**2**	Understanding the various Listeners helps in analyzing and interpreting the action occurs
		PO5	**2**	Understanding the various database access techniques helps in analyzing and interpreting the oops concepts

	CO4	**PO2**	**2**	Applies the knowledge in identifying the appropriate end to end connectivity of DB for reliable communication
		PO4	**3**	Understanding the various end to end protocols helps in analyzing and interpreting the quality of applet.
	CO5	**PO1**	**1**	They could apply the knowledge acquired on various applications over graphics interphase
		PO2	**1**	They could identify the various applications over SQL.
	CO6	**PO4**	**2**	The students could analyze the applications over the Swing layout.
		PO5	**2**	The students could analyze and interpret the applications over the appletviewer.

CO-PSO MAPPING JUSTIFICATION:

CO's	PSO's	JUSTIFICATION
CO1	PSO1	Apply OOPs concept using JAVA for the implementation of DB and frontend backend communication