SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES (AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

UNIT 1: **(9)**

Object Oriented Programming: Basic concepts, Principles, **Program Structure in Java:** Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style.

Data Types, Variables, and Operators :Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final,

Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (- -) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if–else Expressions, Ternary Operator?:, Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, For–Each for Loop, Break Statement, Continue Statement.

UNIT 2: **(9)**

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.
Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT 3: (9)

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three- dimensional Arrays, Arrays as Vectors
Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super

Class- Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and

Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance.

Interfaces: Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT 4: **(9)**

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Auto- boxing and Auto- unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try,

catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.

Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java(Text Book 2)

UNIT5: **(9)**

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.
Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread Priority-Synchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.
Java Database Connectivity: Introduction, JDBC Architecture, Installing MySQL and MySQL Connector/J, JDBC Environment Setup, Establishing JDBC Database Connections, ResultSet Interface
Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events (Text Book 3)

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

Max Marks: 10

S.No.	CO	Questions	BT
		Unit I:	
1	**1**	Explain the different selection statements in Java with suitable example programs.	L3
2	**1**	Illustrate the working of iterative statements in Java with example programs.	L2
3	**1**	Demonstrate the usage of arithmetic, relational, logical, and bitwise operators in Java with example programs.	L4
4	**1**	Explain the various data types in Java with example programs	L2
5	**1**	Differentiate between type casting and type conversion in Java with illustrative example programs	L5
6	**1**	Compare while loop and do-while loop in Java.	L2
7	**1**	Write a Java program to read input using command-line arguments	L2
8	**1**	**Construct a Java program to demonstrate the use of break and continue statements inside nested loops**	L5
9	**1**	Explain the different control statements in Java with suitable example programs.	L4
10	**1**	Explain the different types of operators in Java with suitable example programs.	L3
11	**1**	Explain the basic concepts and principles of Object-Oriented Programming in Java with suitable examples.	**L2**
12	**1**	Explain the concept of operator precedence and associativity in Java with example programs.	**L3**
13	**1**	Illustrate the structure of a simple Java program, explaining the role of tokens, statements, comments, and escape sequences	**L2**
14	**1**	Develop a Java program using a switch statement to display the day of the week based on user input	**L6**

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

15	**1**	Demonstrate with a program the difference between implicit type conversion (widening) and explicit type casting (narrowing) in Java.	**L4**
S.No.	**CO**	**Questions**	**BT**
		Unit II	
1	**2**	Describe the process of declaring a class and creating objects in Java with a simple program	L4
2	**2**	Differentiate between constructors and methods in Java	L3
3	**2**	**Explain the role of the keyword this in Java with example programs**	L3
4	**2**	Explain polymorphism in Java and discuss how it can be achieved through method overloading and method overriding and constructor overloading, with suitable programs.	L4
5	**2**	Explain the use of final keyword with class, method, and variable using example programs	L5
6	**2**	**Explain the concept of functions in Java. Differentiate between parameter passing techniques (pass by value vs. pass by reference) and demonstrate their working with suitable Java programs**	L3
7	**2**	Explain Access Control Mechanism In Java With Suitable Example Programs?	L5
8	**2**	Write A Java Program Assigning One Object To Another Object?	L4
9	**2**	Construct a Java program to calculate the factorial of a number using recursion, and explain how recursion works.	L4
10	**2**	Write a Java program to declare a class and create objects. Demonstrate how to access class members	L3
11	**2**	Construct a Java program with overloaded constructors to initialize student details and display them using methods.	**L6**
12	**2**	Write a Java program to demonstrate method overriding (runtime polymorphism) using inheritance.	**L4**
13	**2**	Develop a Java program to check access control: use private, protected, and public members in different classes and demonstrate accessibility	**L6**

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

S. No.	CO	Questions	BT
		Unit III	
1	**3**	Explain the concept of arrays in Java and illustrate the types of arrays (1D, 2D, 3D) with example programs.	L4
2	**3**	Demonstrate one-dimensional arrays in Java by writing a program to store, access, and print elements.	L3
3	**3**	Write a Java program to demonstrate two-dimensional arrays and explain how elements are stored and accessed.	L3
4	**3**	Develop a Java program to perform binary search on a sorted array and explain the steps	L4
5	**3**	Write a Java program to perform linear search on an array and display the search results.	L5
6	**3**	Explain and demonstrate searching elements in arrays using linear search and binary search with suitable programs.	L3
7	**3**	Construct a Java program to assign one array to another array and explain what happens in memory.	L5
8	**3**	Explain the concept of inheritance in Java and illustrate the types of inheritance (single, multilevel, hierarchical) with example programs.	L4
9	**3**	Write a Java program to demonstrate method overriding and dynamic method dispatch using inheritance	L4
10	**3**	Explain the concept of an interface in Java with a program demonstrating its declaration and implementation.	L3
11	**3**	Write a Java program demonstrating interface inheritance and implementation of extended interfaces.	**L6**
12	**3**	Develop a Java program to calculate the addition of two matrices and display the resulting matrix	**L4**
13	**3**	Write a java program to find max element and min element in arrays.	**L6**
14	**3**	Write a java program to implement sorting array elements (Asc and dec)	L4
15	**3**	Develop a Java program to calculate the multiplication of two matrices and display the resulting matrix	L3

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

S.No.	CO	Questions	BT
		Unit IV:	
1	**4**	Explain the access control modifiers in Java (public, private, protected) with examples.	L4
2	**4**	How does auto-boxing and auto-unboxing work in Java? Give an example.	L3
3	**4**	Explain the purpose of the Formatter class with an example.	L3
4	**4**	What are Temporal Adjuster classes in Java? Give an example of adjusting a date.	L4
5	**4**	How do you import a package and its classes into a Java program?	L5
6	**4**	Write a Java program to implement a user-defined package	L3
7	**4**	Write a Java program to demonstrate auto-boxing and auto-unboxing using wrapper classes.	L5
8	**4**	Write a Java program to format output using the Formatter class.	L4
9	**4**	Write a Java program using Temporal Adjusters to find the next Monday from today.	L4
10	**4**	Demonstrate access control using public, private, and protected in a Java program	L3
11	**4**	Explain the purpose of try, catch, throw, throws, and finally blocks.	**L6**
12	**4**	Demonstrate multiple catch blocks in Java with an example.	**L4**
13	**4**	Analyze the differences between throw and throws in Java.	**L6**
14	**4**	Write a Java program to read data from a file using FileReader.	L4
15	**4**	Write a Java program to write data into a file using FileWriter.	L3

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

S.No.	CO	Questions	BT
		Unit V:	
1	**5**	Define string handling in Java and list the classes used for string manipulation.	L4
2	**5**	What is multithreading? Explain the need for multithreaded programming.	L3
3	**5**	**What is JDBC and what are its components?**	L3
4	**5**	Explain the architecture of JDBC with a diagram.	L4
5	**5**	Write a Java program demonstrating thread synchronization using the synchronized keyword	L5
6	**5**	Analyze a JDBC program that executes multiple SQL queries. Explain the role of `ResultSet` in fetching data.	L3
7	**5**	Analyze the difference in performance between using `String`, `StringBuffer`, and `StringBuilder` in repetitive string concatenation.	L5
8	**5**	What are deadlocks and race conditions? Explain with examples.	L4
9	**5**	Explain the structure of a JavaFX application window.	L4
10	**5**	What string and types of string functions with example programs in java ?	L3
11	**5**	Explain string handling mechanism in java with example program?	**L6**
12	**5**	Explain thread life cycle in java	**L4**
13	**5**	Explain inter thread communication in java(hint :sleep(),wait(),notify()etc	**L6**
14	**5**	Implement a java program how to connect with JDBC	L4

Note: L1-Remembering, L2-Understanding, L3-Applying, L4-Analyzing, L5-Evaluating, and L6-Creating

SREENIVASA INSTITUTE OF TECHNOLOGY AND MANAGEMENT STUDIES
(AUTONOMOUS)

QUESTION BANK

Year / Semester: II B.Tech III Semester **Regulation: R23**

Subject and Code: OBJECT ORIENTED PROGRAMMING THROUGH JAVA and 23CSE232T

Instruction to Faculty Members:

The Six Levels of Bloom's Taxonomy:

1. **Remembering:** Retrieving, recognizing, and recalling relevant knowledge from long-term memory (e.g., list, define, name, locate).
2. **Understanding:** Constructing meaning, explaining ideas, or concepts (e.g., summarize, interpret, classify, compare).
3. **Applying:** Using information in new situations or implementing procedures to solve problems (e.g., solve, use, demonstrate, implement).
4. **Analyzing:** Breaking material into constituent parts, determining how the parts relate to one another and to an overall structure (e.g., contrast, categorize, distinguish, diagram).
5. **Evaluating:** Making judgments based on criteria and standards through checking and critiquing (e.g., judge, critique, justify, defend, argue).
6. **Creating:** Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure (e.g., design, construct, develop, formulate).