

## II YEAR III SEMESTER

### OBJECT ORIENTED PROGRAMMING USING JAVA

#### Course Objectives

As the business environment becomes more sophisticated, the software development (software engineering is about managing complexity) is becoming increasingly complex. As of the best programming paradigm which helps to eliminate complexity of large projects, Object Oriented Programming (OOP) has become the predominant technique for writing software in the past decade. Many other important software development techniques are based upon the fundamental ideas captured by object-oriented programming.

#### Course Outcomes

At the end of this course student will:

1. Understand the concept and underlying principles of Object-Oriented Programming
2. Understand how object-oriented concepts are incorporated into the Java programming language
3. Develop problem-solving and programming skills using OOP concept
4. Understand the benefits of a well structured program
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java
6. Develop efficient Java applets and applications using OOP concept
7. Become familiar with the fundamentals and acquire programming skills in the Java language.

#### UNIT-1

**FUNDAMENTALS OF OBJECT – ORIENTED PROGRAMMING** :Introduction, Object Oriented paradigm, Basic Concepts of OOP, Benefits of OOP, Applications of OOP, Java features: **OVERVIEW OF JAVA LANGUAGE**: Introduction, Simple Java program structure, Java tokens, Java Statements, Implementing a Java Program, Java Virtual Machine, Command line arguments. **CONSTANTS, VARIABLES & DATA TYPES**: Introduction, Constants, Variables, Data Types, Declaration of Variables, Giving Value to Variables, Scope of variables, Symbolic Constants, Type casting, Getting Value of Variables, Standard Default values; **OPERATORS & EXPRESSIONS**.

#### UNIT-II

**DECISION MAKING & BRANCHING**: Introduction, Decision making with if statement, Simple if statement, if. Else statement, Nesting of if. else statements, the else if ladder, the switch statement, the conditional operator. **LOOPING**: Introduction, The While statement, the do-while statement, the for statement, Jumps in loops.

**CLASSES, OBJECTS & METHODS:** Introduction, Defining a class, Adding variables, Adding methods, Creating objects, Accessing class members, Constructors, Method overloading, Static members, Nesting of methods;

### **UNIT-III**

**INHERITANCE:** Extending a class, Overloading methods, Final variables and methods, Final classes, Abstract methods and classes;

**ARRAYS, STRINGS AND VECTORS:** Arrays, One-dimensional arrays, Creating an array, Two – dimensional arrays, Strings, Vectors, Wrapper classes;

**INTERFACES: MULTIPLE INHERITANCE:** Introduction, Defining interfaces, Extending interfaces, Implementing interfaces, Assessing interface variables;

### **UNIT-IV**

**MULTITHREADED PROGRAMMING:** Introduction, Creating Threads, Extending the Threads, Stopping and Blocking a Thread, Lifecycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the ‘Runnable’ Interface.

**MANAGING ERRORS AND EXCEPTIONS:** Types of errors : Compile-time errors, Runtime errors, Exceptions, Exception handling, Multiple Catch Statements, Using finally statement,

### **UNIT-V**

**APPLET PROGRAMMING:** local and remote applets, Applets and Applications, Building Applet code, Applet Life cycle: Initialization state, Running state, Idle or stopped state, Dead state, Display state.

**PACKAGES:** Introduction, Java API Packages, Using System Packages, Naming conventions, Creating Packages, Accessing a Package, using a Package.

**MANAGING INPUT/OUTPUT FILES IN JAVA:** Introduction, Concept of Streams, Stream classes, Byte Stream Classes, Input Stream Classes, Output Stream Classes, Character Stream classes: Reader stream classes, Writer Stream classes, Using Streams, Reading and writing files.

### **Reference Books:**

1. E.Balaguruswamy, Programming with JAVA, A primer, 3e, TATA McGraw-Hill Company.

2. John R. Hubbard, Programming with Java, Second Edition, Schaum's outline Series, TATA McGraw-Hill Company.
3. Deitel & Deitel. Java TM: How to Program, PHI (2007)
4. Java Programming: From Problem Analysis to Program Design- D.S Mallik
5. Object Oriented Programming Through Java by P. Radha Krishna, Universities Press (2008)

**Student Activity:**

- 1. Create a front end using JAVA for the student database created**
- 2. Learn the difference between ODBC and JDBC**

**OBJECT ORIENTED PROGRAMMING USING JAVA LAB**

1. Write a program to perform various String Operations
2. Write a program on class and object in java
3. Write a program to illustrate Function Overloading & Function Overriding methods in Java
4. Write a program to illustrate the implementation of abstract class
5. Write a program to implement Exception handling
6. Write a program to create packages in Java
7. Write a program on interface in java
8. Write a program to Create Multiple Threads in Java
9. Write a program to Write Applets to draw the various polygons
10. Write a program which illustrates the implementation of multiple Inheritance using interfaces in Java
11. Write a program to assign priorities to threads in java

Andhra Pradesh State Council of Higher Education  
**B.Sc Computer Science /Information Technology Under CBCS with effect from  
Academic Year 2015-2016  
Revised in April, 2016**

---

**Code No:**

**SRI VENKATESWARA UNIVERSITY: TIRUPATI**

**B.Sc(CBCS)**

**III SEMESTER EXAMINATIONS**

OBJECT ORIENTED PROGRAMMING USING JAVA

**Time: 3 Hours**

**Max. Marks: 75**

**Note:** This question paper contains two parts A and B.

Part A is compulsory which carries 25 marks. Answer any five of the following questions in Part A.

Part B consists of 5 Units. Answer any one full question from each unit.

Each question carries 10 marks

**PART - A**

**Answer any Five of the following. All questions carry equal marks**

**5 x 5 = 25 Marks**

1. Write about object oriented paradigm.
2. Give a note on data types in java language
3. Describe the conditional operator with example?
4. Write about the method of accessing class members?
5. Write about Final classes in inheritance.
6. What is the concept of wrapper classes?
7. Describe the process of creating threads?
8. Write about the process of creating packages

**PART - B**

**Answer one question from each Unit. All questions carry equal marks**

**5 x 10 = 50 Marks**

**UNIT - I**

9. Explain the basic concepts of OOP with Benefits and Applications of OOP?

**OR**

- 10 (a) Write about command line arguments  
(b) Write about Scope of variables and Type casting in Java.

**UNIT-II**

11. Explain the Decision Making statement & write a program which will display message "Success" if a particular value is greater than 5. It then displays a message "Executed successfully" and complete its execution.

**OR**

12. Explain the process of Method overloading with example?

**UNIT-III**

13. What is an Array? Explain in detail about one dimensional and two dimensional arrays with an example?

**OR**

14. Explain the concept of multiple inheritance with an illustration

**UNIT-IV**

15. (a) Explain the Lifecycle of a Thread?  
(b) Explain the types of errors with example.

**OR**

16. Write about Multiple catch block program?

**UNIT - V**

17. Explain in detail about Applet life cycle and its applications?

**OR**

18. Explain the Concept of streams with detail example?

## II YEAR IV SEMESTER

### DATA STRUCTURES

#### Course Objectives

To introduce the fundamental concept of data structures and to emphasize the importance of data structures in developing and implementing efficient algorithms..

#### Course Outcomes

After completing this course satisfactorily, a student will be able to:

1. Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
2. Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs.
3. Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
4. Demonstrate different methods for traversing trees
5. Compare alternative implementations of data structures with respect to performance
6. Compare and contrast the benefits of dynamic and static data structures implementations
7. Describe the concept of recursion, give examples of its use, describe how it can be implemented using a stack .
8. Discuss the computational efficiency of the principal algorithms for sorting, searching, and hashing.

#### UNIT I

**Concept of Abstract Data Types (ADTs)-** Data Types, Data Structures, Storage Structures, and File Structures, Primitive and Non-primitive Data Structures, Linear and Non-linear Data Structures.

**Linear Lists** – ADT, Array and Linked representations, Pointers.

**Arrays** – ADT, Mappings, Representations, Sparse Matrices, Sets – ADT, Operations

**Linked Lists:** Single Linked List, Double Linked List, Circular Linked List , applications

#### UNIT II

**Stacks:** Definition, ADT, Array and Linked representations, Implementations and Applications

**Queues:** Definition, ADT, Array and Linked representations, Circular Queues, Dequeues, Priority Queues, Implementations and Applications.

#### UNIT III

**Trees:** Binary Tree, Definition, Properties, ADT, Array and Linked representations, Implementations and Applications. Binary Search Trees (BST) – Definition, ADT, Operations and Implementations, BST Applications. Threaded Binary Trees, Heap trees.

**UNIT IV**

**Graphs** – Graph and its Representation, Graph Traversals, Connected Components, Basic Searching Techniques, Minimal Spanning Trees

**UNIT- V**

**Sorting and Searching:** Selection, Insertion, Bubble, Merge, Quick, Heap sort, Sequential and Binary Searching.

**REFERENCE BOOKS**

1. D S Malik, Data Structures Using C++, Thomson, India Edition 2006.
2. Sahni S, Data Structures, Algorithms and Applications in C++, McGraw-Hill, 2002.
3. SamantaD, Classic Data Structures, Prentice-Hall of India, 2001.
4. Heilman G I, Data Structures and Algorithms with Object-Oriented Programming, Tata McGraw-Hill. 2002. (Chapters I and 14).
5. Tremblay P, and Sorenson P G, Introduction to Data Structures with Applications, Tata McGraw-Hill,

**Student activity:**

1. Create a visible stack using C-graphics
2. Create a visible Queue using C-graphics

**DATA STRUCTURES USING JAVA LAB**

1. Write a Program to implement the Linked List operations
2. Write a Program to implement the Stack operations using an array.
3. Write Programs to implement the Queue operations using an array.
4. Write Programs to implement the Stack operations using a singly linked list.
5. Write Programs to implement the Queue operations using a singly linked list.
6. Write a program for arithmetic expression evaluation
7. Write a program to implement Double Ended Queue using a doubly linked list.
8. Write a program to search an item in a given list using Linear Search and Binary Search
9. Write a program for Quick Sort
10. Write a program for Merge Sort
11. Write a program on Binary Search Tree operations(insertion, deletion and traversals)
12. Write a program for Graph traversals