

CO OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Rationale:

Java is a computer programming language that is concurrent, class-based, and object-oriented, It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java is one of the most popular programming languages in use, particularly for client-server web applications, The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. This subject provides a student an insight to understand and implement the OOP concepts, do applet, graphics and multithreaded programming and Interact with the files. It also builds strong foundation for advanced java programming.

TEACHING AND EXAMINATION SCHEME:

Teaching Schedule Per Week			Progressive Assessment		Examination Schedule (Marks)		
Lectures	Practical	Tutorials	Test	TW	Theory	Practical Exam	Total
3	2	-	25	25	75	50	175

DETAILED COURSE CONTENTS:

UNIT 1: INTRODUCTION TO JAVA

(15 Marks)

- 1.1 Basic concept of object-oriented programming
 - 1.1.1 Objects & classes
 - 1.1.2 Data abstraction & encapsulation
 - 1.1.3 Inheritance
 - 1.1.4 Polymorphism
 - 1.1.5 Dynamic binding
 - 1.1.6 Message communication
 - 1.1.7 Advantages & applications of OOP.
- 1.2 Java features
 - 1.2.1 Compiled & interpreted
 - 1.2.2 Platform independent & portable
 - 1.2.3 Object oriented
 - 1.2.4 Robust & secure
 - 1.2.5 Distributed
 - 1.2.6 Simple, small & familiar
 - 1.2.7 Multithreaded & interactive
 - 1.2.8 High performance
 - 1.2.9 Dynamic & extensible
- 1.3 Java Environment
- 1.4 Overview of Java language
 - 1.4.1 Java program structure
 - 1.4.2 Tokens
 - 1.4.3 Java statements
- 1.5 Constants, variables & data types
 - 1.5.1 Constants
 - 1.5.2 Variables
 - 1.5.3 Data types
 - 1.5.4 Declaration of variables
 - 1.5.5 Giving values to variables

- 1.5.6 Scope of variables
- 1.5.7 Symbolic constants
- 1.5.8 Type casting
- 1.5.9 Standard default values
- 1.6 Operators & expressions
 - 1.6.1 Arithmetic operators
 - 1.6.2 Relational
 - 1.6.3 Logical operators
 - 1.6.4 Assignment operators
 - 1.6.5 Increment/decrement operators
 - 1.6.6 Conditional operators
 - 1.6.7 Bitwise operators
 - 1.6.8 Special operators
 - 1.6.9 Arithmetic expressions
 - 1.6.10 Evaluation of expression
 - 1.6.11 Precedence of arithmetic operators
 - 1.6.12 Type conversion in expression
 - 1.6.13 Operator precedence & associativity
- 1.7 Decision making, branching & looping
 - 1.7.1 If statement
 - 1.7.2 If-else, nested if-else, if-else if ladder
 - 1.7.3 Switch
 - 1.7.4 While
 - 1.7.5 Do-while
 - 1.7.6 For
 - 1.7.7 Jumps in loops (break, continue)
 - 1.7.8 Labeled loop
 - 1.7.9 Nested loops

UNIT 2: CLASSES, OBJECTS & METHODS

(15 Marks)

- 2.1 Classes & objects
 - 2.1.1 Introduction
 - 2.1.2 Defining a class
 - 2.1.3 Field declaration
 - 2.1.4 Method declaration
 - 2.1.5 Creating objects
 - 2.1.6 Accessing class members
 - 2.1.7 Constructors
 - 2.1.8 Method overloading
 - 2.1.9 Static methods
 - 2.1.10 Nesting of methods
- 2.2 Inheritance
 - 2.2.1 Defining a subclass
 - 2.2.3 Subclass constructor
 - 2.2.4 Multilevel inheritance
 - 2.2.5 Hierarchical inheritance
 - 2.2.6 Overriding methods
 - 2.2.7 Final variables & methods
 - 2.2.8 Final classes
 - 2.2.9 Finalizer method
 - 2.2.10 Abstract methods & classes
- 2.3 Visibility control

UNIT 3: ARRAYS, INTERFACES & PACKAGES

(15 Marks)

- 3.1 Arrays, Strings & Vectors
 - 3.1.1 One dimensional array
 - 3.1.2 Creating an array
 - 3.1.3 Two dimensional array
 - 3.1.4 Strings
 - 3.1.4.1 String array
 - 3.1.4.2 String methods
 - 3.1.4.3 String buffer class
 - 3.1.5 Vectors
 - 3.1.6 Wrapper classes
 - 3.1.7 Enumerated types
- 3.2 Interfaces
 - 3.2.1 Introduction
 - 3.2.2 Defining interfaces
 - 3.2.3 Extending interfaces
 - 3.2.4 Implementing interfaces
 - 3.2.5 Accessing interface variables
- 3.3 Packages
 - 3.3.1 Introduction
 - 3.3.2 Java API packages
 - 3.3.3 Using system packages
 - 3.3.4 Naming conventions
 - 3.3.5 Creating packages
 - 3.3.6 Accessing a package
 - 3.3.7 Adding a class to a package
 - 3.3.8 Hiding classes

UNIT 4: MULTITHREADING & EXCEPTION HANDLING

(15 Marks)

- 4.1 Multithreaded Programming
 - 4.1.1 Creating threads
 - 4.1.2 Extending the thread class
 - 4.1.3 Stopping & Blocking the thread
 - 4.1.4 Lifecycle of a thread
 - 4.1.5 Using thread methods
 - 4.1.6 Thread exceptions
 - 4.1.7 Thread priority
 - 4.1.8 Synchronization
 - 4.1.9 Implementing the Runnable interface
- 4.2 Exception handling
 - 4.2.1 Types of errors
 - 4.2.2 Exceptions
 - 4.2.3 Syntax of exception handling code
 - 4.2.4 Multiple catch statements
 - 4.2.5 Using finally statements
 - 4.2.6 Throwing our own exception

UNIT 5: APPLET & GRAHICS

(15 Marks)

- 5.1 Applet Programming
 - 5.1.1 Introduction
 - 5.1.2 Applet lifecycle
 - 5.1.3 Building Applet code
 - 5.1.4 Creating an executable Applet
 - 5.1.5 Designing a webpage
 - 5.1.6 Applet tag

- 5.1.7 Adding Applet to a HTML file
- 5.1.8 Running the Applet
- 5.1.9 Passing parameters to Applet
- 5.1.10 Aligning the display
- 5.2 Graphics Programming
 - 5.2.1 Graphics class
 - 5.2.2 Lines & rectangles
 - 5.2.3 Circles & ellipses
 - 5.2.4 Drawing arcs
 - 5.2.5 Drawing polygon

Text Book:

1. Programming in Java- E Balagurusamy

Reference Book:

1. Timothy Budo, "An Introduction to Object-Oriented Programming with Java", Pearson Education, Latest Edition.

Suggested List of practicals:

1. Program to implement constants, variables, operators and expressions
2. Program to Implement if-else
3. Program to implement loops
4. Program to implement switch-case
5. Program to implement arrays and strings
6. Program to implement packages
7. Program to implement interfaces
8. Program to implement multithreading
9. Program to implement exception handling
10. Program to implement applets and graphics

