1.1 Aims and Objectives

1.2 Introduction

1.3 Procedural Programming Language and Object Oriented Language

1.3.1 Structure of Procedural Programming Language

1.3.2 Object Oriented Language

1.4 Introduction to OOP

1.4.1 Object Oriented Programming Paradigm

1.5 Concepts of Object Oriented Programming

1.6 Applications of OOP

1.7 Advantages of OOP

1.8 History of Java

1.8.1 Introduction

1.8.2 Evolution

1.8.3 Some Features that make Java better than C

1.8.3 Hardware and Software Requirements

1.9 Features or BUZZ Words

1.10 Java Virtual Machine

1.10.1 JVM Architecture

1.11 Java Applications

1.12 Program Structure

1.12.1 Developing a Simple Java Application

1.12.2 Anatomy of the Application Program

1.13 Java Building Elements

1.13.1 Variables

1.13.2 Constants

1.13.3 Separators

1.13.4 Expressions

1.13.5 Data Types

1.13.6 Identifiers

1.13.7 Literals

1.13.8 Operators

1.13.8.1 Types of Operators

1.14 Operator Precedence and Associativity

1.15 Primitive Type Conversion and Casting

1.15.1 Automatic Conversion (implicit conversion)

1.15.2 Casting Incompatible Types

1.16 Control Flow Statements

1.16.1 Selection Statements

1.16.1.1 If Statement

1.16.1.2 Switch Statement

1.16.2 Iteration Statements

1.16.3 Jump Statements

1.17 Classes and Objects

1.17.1 Class

1.17.2 Object

1.17.3 Passing Objects as Parameters

1.18 Methods

1.18.2 Calling Methods

1.18.3 Returning a Value

1.18.4 Adding a Method that Takes the Parameters

1.19 Method Overloading

1.19.1 Creating Methods

1.20 Outcomes

1.21 Summary

1.22 Review Questions

1.23 Review Programs

1.24 Practice Questions

1.25 Multiple Choice Questions

Ch 2

2.0 Aim and objectives

2.1 Constructors

2.1.1 Types of Constructors

2.1.2 Constructor Overloading

2.1.3 Copy Constructor

2.2 Garbage Collection

2.3 Importance of static Keyword

2.3.1 Variables

2.3.2 Static Methods

2.3.3 Static Block

2.4 The this Keyword

2.5 Arrays

2.5.1 Creating an Array

2.5.2 Declaring a Variable to Refer to an Array

2.5.3 Creation of an Array and Declaration of a Variable to Refer to an Aray

2.5.4 Accessing an Array Element

2.5.5 Array Initializers

2.5.6 One Dimensional Array

2.5.7 Initializing a Two-Dimensional Array

2.5.8 Multidimensional Arrays

2.5.9 Alternative Array Declaration Syntax

2.5.10 Arrays of Objects

2.5.11 Array of Arrays

2.5.12 Copying Arrays

2.5.13 Sorting Array Elements

2.5.14 Searching an Array

2.6 Command Line arguments

2.7 Nested Classes

2.7.1 Types of Nested Classes

2.7.2 Static Nested Classes

2.7.3 Non-static Nested Classes (Inner Classes)

2.8 Inheritance

2.8.1 Introduction

2.8.2 Sub class and Super class

2.8.3 Types of Inheritance

2.8.4 Member Access and Inheritance

2.8.5 Super Class Variable and Sub Class Object

2.8.6 Subtype, Subclasses and Substitutability

2.8.7 Forms of Inheritance

2.8.8 The Benefits of Inheritance

2.8.9 Creating a Multilevel Hierarchy

2.8.10 The uses of the “super” keyword

2.8.11 Calling Constructor

2.8.12 Using final with Inheritance

2.9 Polymorphism

2.9.1 Introduction

2.9.2 Dynamic Method Dispatch

2.9.3 Method Overriding

2.9.4 Difference between Overloading and Overriding

2.10 Using Abstract Classes

2.11 Object Classes

2.12 Wrapper Classes

2.13 Outcomes

2.14 Summary

2.15 Review Questions

2.16 Programs Related to the Topics Discussed in this Unit

2.17 Practice Questions

2.18 Multiple Choice Questions

Chapter-3: Packages

3.1 Aims and Objectives

3.2 Introduction to Packages

3.3 Predefined Packages (or) JAVA API Packages

3.3.1 Using the System Packages

3.4 Creating a package

3.4.1 Naming a Package

3.5Using Package Members

3.5.1 Referring to a Package Member by Name

3.5.2 Importing a Package Member

3.5.3 Importing an Entire Package

3.6 Access Protection

3.7 Importing Classes from Other Packages

3.8 Benefits of Packages

3.9 Path and Classpath

3.9.1 Update the Path Environment Variable

3.9.2 Update the Path Variable (Solaris and Linux)

3.9.3 Checking the Classpath variable (All platforms)

3.9.4 Difference between Path and Classpath

3.10 Important Packages

3.11 Outcomes

3.12 Summary

3.13 Review Questions

3.14 Multiple Choice Questions

Chapter-4: Interfaces

4.0 Aims and Objectives

4.1 Introduction to Interfaces

4.2 Defining Interface

4.3 Implementing the interface

4.4 Nested Interfaces

4.4.1 Interface in a Class

4.4.2 Interface in another Interface

4.5 Extending the Interfaces

4.6 Java Interface Variables

4.7 Java Interface References

4.8 Difference between abstract class and interface

4.9 Applications of Interfaces in Java

4.10 Extends VS. Implements

4.11 Outcomes

4.12 Summary

4.13 Review Questions

4.14 Practice Questions

4.15 Multiple Choice Questions

Chapter-5: Exception Handling

5.0 Aims and Objectives

5.1 Introduction

5.2 Exception Handling in Java

5.2.1 What is Exception in Java

5.3 Fundamentals of Exception Handling

5.3.1 Advantages of Exception Handling

5.3.2 Hierarchy of Java Exception Classes

5.3.3 Types of Java Exceptions

5.4 Uncaught Exceptions

5.5 Java Exception Keywords

5.6 Using Try and Catch Blocks

5.7 Java Multi Catch Block

5.8 Java Nested Try Block

5.9 Java Finally Block

5.9.1 Difference Between Final, Finally and Finalize in Java

5.10 Java throw Exception

5.11 Using Java throws Keyword

5.11.1 Difference between throw and throws in Java

5.12 Java Built in Exceptions

5.12.1 Following is the list of Java Unchecked Runtime Exception.

5.12.2 Following is the list of Java Checked Exceptions Defined in java.lang.

5.13 User Defined Exceptions

5.13.1 Exceptions Methods

5.14 Outcomes

5.15 Summary

5.16 Review Questions

5.17 Practice Questions

5.18 Multiple Choice Questions

Ch 6

6.0 Aims and Objectives

6.1.Introduction

6.2 Difference between Multiprocessing and Multithreading

6.3 What is a Thread

6.3.1 Package Support of Threads

6.3.2 The Java Thread Model

6.3.4 Advantages of the Multithreading

6.3.5 States of the Thread

6.4. Life Cycle of a Thread

6.5. The Thread class and Runnable Interface

6.6 The Main Thread

6.7 Creation of Threads

6.7.1 Implementing the Runnable Interface

6.7.2 Extending the Thread Class

6.8 Creating Multiple Threads

6.9 When a Thread is ended

6.10Thread Priorities

6.11Synchronization

6.12 Inter-Thread Communication

6.14Suspending, Blocking and Stopping Threads

6.15Thread Expections

6.16Thread Group

6.16.1 Class Declaration

6.16.2 Constructors of Thread Group Class

6.16.3 Methods of Thread Group Class

6.17 Thread Class

6.17.1 Class Declaration

6.17.2 Fields

6.17.3 Class Constructors

6.17.3 Methods and its Description

6.18 Outcomes

6.19. Summary

6.20 Review Questions

6.21 Multiple Choice Questions

Ch-7

7.0 Aims and Objectives

7.1 Introduction

7.1.1 Files in Java

7.2 Streams

7.2.1 Key Features of Streams

7.2.2 Byte and Character Streams

7.2.3 Using the Streams

7.3 Java Input and Output

7.3.1 Input Stream

7.3.2 Output Stream

7.4 InputStream Class Methods and OutputStream Class Methods

7.4.1 InputStream Class Methods

7.4.2 OutputStream Class Methods

7.5 Byte Streams and Character Streams

7.5.1 Byte Streams

7.5.1.1 FileInputStream Class

7.5.1.2 FileOutputStream Class

7.5.2 Character Streams

7.6 Standard Streams

7.7 Reading Ordinary Text Files in Java

7.8 Reading Binary Files in Java

7.9 Writing Text Files in Java

7.10 Writing Binary Files in Java

7.11 Random Access Files

7.11.1 Datalnput and Data Output Interfaces

7.12 Outcomes

7.13 Summary

7.14 Review Questions

7.15 Multiple Choice Questions

Chpater-8: Applets

8.0 Aims and Objectives

8.1 Introduction

8.1.1 Basic Differences between an Application and Applet

8.2 Applet Basics

8.3 Applet Architecture

8.4 Life Cycle of an Applet

8.5 The Applet Class

8.5.1 Building Applet Code

8.5.2 Creating an Executable Applet

8.5.3 Methods of Applet Class

8.6 Viewing Applets

8.6.1 < Applet> Tag

8.6.2 About the < Applet> Tag

8.6.3 List of Attributes that can be used with < Applet > Tag

8.7 Requesting the Repaint() Method

8.8 Using the Status Window

8.9 Passing Parameters to Applet

8.10 Getting the Input from the User

8.11 Events

8.11.1 Event Handling

8.11.2 Types of Event Handling Mechanisms

8.12 The Delegation Event Model

8.12.1 What is an Event?

8.12.2 Event Sources

8.12.3 Event Listeners

8.12.4 The Event Classes

8.12.5 Event Listener Interfaces

8.13 Using the Delegation Model for Handling the Mouse Events

8.14 Keyboard Events

8.15 Adapter Classes

8.16 Inner Classes

8.17 Outcomes

8.18 Summary

8.19 Review Questions

8.20 Programs Related to the Topics Discussed in this Unit

8.21 Multiple Choice Questions