CSE380:PROGRAMMING IN JAVA

L:0 T:0 P:4 Credits:2

Course Outcomes: Through this course students should be able to

CO1 :: evaluate the Java platform's ability to achieve platform independence and differentiate between various Java platforms and their uses.

CO2 :: apply string manipulation, data formatting, and conditionals effectively in Java programs.

CO3 :: describe objects and classes, and demonstrate the use of methods, encapsulation, inheritance, and polymorphism in Java.

CO4 :: analyze access levels, method overriding, and abstract classes to design robust Java applications.

CO5 :: create modular Java programs by implementing interfaces, utilizing lambda expressions, and effectively handling exceptions and assertions.

CO6 :: demonstrate I/O fundamentals and utilize collections and generics to manage data efficiently in Java applications.

List of Practicals / Experiments:

Java Platform Overview

- Defining how the Java language achieves platform independence,Differentiating between the Java ME, Java SE, and Java EE Platforms, Evaluating Java libraries,middle-ware, and database options, Defining how the Java language continues to evolve

What Is a Java Program?

- Introduction to Computer Programs, Key Features of the Java Language, The Java Technology and Development Environment, Running/testing a Java program

Creating a Java Main Class

- Java Classes, The main Method

Data In the Cart

- Introducing variables, Working with Strings, Working with numbers,Manipulating numeric data

Managing Multiple Items

- Working with Conditions, Working with a List of Items, Processing alist of items

Manipulating and Formatting the Data in Your Program

- Using the String Class, Using theStringBuilder Class, More about primitive data types, The remaining numeric operators, Promotingand casting variables

More on Conditionals

- Relational and conditional operators, Ways to use if/else constructs, Using Switch Statements.

More on Arrays and Loops

- Working with Dates, Parsing the args Array, 2D Arrays, Alternate Looping Constructs, Nesting Loops, The ArrayList class.

Describing Objects and Classes

- Working with objects and classes, Defining fields and methods, Declaring, Instantiating, and Initializing Objects, Working with Object References, Doing more with Arrays.

Creating and Using Methods

- Using methods, Method arguments and return values, Static methods and variables, How Arguments are Passed to a Method, Overloading a method.

Using Inheritance

- Overview of inheritance, Working with sub-classes and super-classes, Overriding methods in the superclass, Introducing polymorphism, Creating and extending abstract classes, Modeling business problems using Java classes, Making classes immutable.

Overriding Methods, Polymorphism, and Static Classes

- Using access levels: private, protected, default, and public, Overriding methods, Using virtual method invocation, Using varargs to specify variable arguments, Using the instance of operator to compare object types, Using upward and downward casts, Modeling business problems using the static keyword, Implementing the singleton design pattern.

Abstract and Nested Classes

- Designing general-purpose base classes by using abstract classes, Constructing abstract Java classes and subclasses, Applying final keyword in Java, Distinguish between top-level and nested classes.

Using Interfaces

- Polymorphism in the JDK foundation classes, Using Interfaces, Using the List Interface, Introducing Lambda expressions.

Interfaces and Lambda Expressions

- Defining a Java interface, Choosing between interface, inheritance and class inheritance, Extending an interface, Defaulting methods, Anonymous inner classes, Defining a Lambda Expression.

Exceptions and Assertions

- Handling Exceptions: An overview, Propagation of exceptions, Catching and throwing exceptions, Handling multiple exceptions and errors, Defining the purpose of Java exceptions, Using the try and throw statements, Using the catch, multi-catch, and finally clauses, Autoclose resources with a try-with-resources statement, Recognizing common exception classes and categories, Creating custom exceptions, Testing invariants by using assertions.

I/O Fundamentals

- Describing the basics of input and output in Java, Read and write data from different sources, Using streams to read and write files, Writing and read objects using serialization.

Collections and Generics

- Creating a custom generic class, Using the type inference diamond to create an object, Creating a collection by using generics, Implementing an ArrayList, Implementing a TreeSet, Implementing a HashMap, Implementing a Deque, Ordering collections.

Text Books:

1. INTRODUCTION TO JAVA PROGRAMMING by Y. DANIEL LIANG, PEARSON

References:

1. JAVA THE COMPLETE REFERENCE by HERBERT SCHILDT, MCGRAW, HILL & CO PUB
2. HEAD FIRST JAVA: A BRAIN-FRIENDLY GUIDE by KATHY SIERRA, O Reilly Publishing