

# CSE310: PROGRAMMING IN JAVA

L:3 T:0 P:2 Credits:4

**Course Outcomes:** Through this course students should be able to

- Write JAVA code that uses tokens, literals, operators and constructs
- Understand the accessibility of fields and methods of an object and use of String and StringBuilder classes
- Apply the concept of inheritance to reuse and extend the features of existing class with access control
- Analyse the need of implementing Interfaces for multiple inheritance and Lambda Expressions
- Demonstrate Object Serialization with File handling and Exception handling to deal with run-time errors
- Differentiate among different types of collections by implementing their methods

## Unit I

**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 a list of items

## Unit II

**Manipulating and Formatting the Data in Your Program:** Using the String Class, Using the StringBuilder Class, More about primitive data types, The remaining numeric operators, Promoting and casting variables

**More on Conditionals:** Relational and conditional operators, More ways to use if/else constructs, Using Switch Statements

**More on Arrays and Loops:** Working with Dates, Parsing the args Array, Two-dimensional 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

## Unit III

**Using Encapsulation:** Access Control, Encapsulation, Overloading constructors

**Using Inheritance:** Overview of inheritance, Working with subclasses and superclasses, 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 instanceof operator to compare object types, Using upward and downward casts, Modeling business problems using the static keyword, Implementing the singleton design pattern

## Unit IV

**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

## Unit V

**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 the console, Using streams to read and write files, Writing and read objects using serialization

## Unit VI

**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

### List of Practical:

**Creating a Java Main Class:** Program to implement a java class

**Managing Multiple Items:** Program to demonstrate the use of list of items

**Manipulating and Formatting the Data in Your Program:** Program to demonstrate the uses of String and StringBuilder

**Describing Objects and Classes:** Program to demonstrate the instantiation of class and accessing the attributes using object of class

**Using Inheritance:** Program to demonstrate the inheritance and its importance

**Overriding Methods, Polymorphism, and Static Classes:** Program to implement polymorphism and using proper access control

**Abstract and Nested Classes:** Program to demonstrate the use of abstract class and nested class

**Interfaces and Lambda Expressions:** Program to demonstrate the inheritance through interfaces and use of Lambda Expressions

**Exceptions and Assertions:** Program to demonstrate the use of all the keywords used for exception handling and need of assertion

**I/O Fundamentals:** Program to implement read and write operation using console and File

**Generics:** Program to define generic class and creating generic collection

**Collections:** Program to implement ArrayList, HashMap, TreeSet and Deque

### References:

1. INTRODUCTION TO JAVA PROGRAMMING by Y. DANIEL LIANG, PEARSON
2. JAVA THE COMPLETE REFERENCE by HERBERT SCHILDT, MCGRAW HILL EDUCATION