Java8F - *Java SE 8 Fundamentals*

DURATION: 5 Days; Instructor-led

WHAT YOU WILL LEARN
This Java SE 8 Fundamentals training introduces you to object-oriented programming using the Java language. Through hands-on exercises, you'll begin to build a baseline of knowledge to propel your career in development.

After completing this course, students will be able to:
- Use Java programming language constructs to create a Java technology application.
- Use decision and looping constructs and methods to dictate program flow.
- Understand basic object oriented concepts such as inheritance, encapsulation, and abstraction.
- Use and manipulate object references, and to write simple error handling code.
- Use the new SE 8 java.time and java.time.format packages to format and print the local date and time.
- Specify a data modification by passing a predicate lambda expression to the Collections class.

AUDIENCE
This course is aimed at beginners to programming that wish to learn the Java language. From administrator, developer, implementer, systems administrator, manager with a non-technical and non-programming background.

PREREQUISITES
Familiarization with programming concepts is useful but is not mandatory.

The course materials, lectures, and lab exercises are in English. To benefit fully from the instruction, students need an understanding of the English language.

OBJECTIVES
- Write Java code that uses variables, arrays, conditional and loop constructs
- Manipulate primitive numeric data and string data using Java operators
- Create Java classes and use object references
- Access the fields and methods of an object
- Manipulate text data using the methods of the String and StringBuilder classe
- Use casting without losing precision or causing errors
- Declare, override, and invoke methods
- Access and create static fields and methods
- Use classes from the java.time and java.time.format packages to format and print the local date and time
- Encapsulate a class using access modifiers and overloaded constructors
- Define and implement a simple class hierarchy

- Demonstrate polymorphism by implementing a Java Interface
- Use a Predicate Lambda expression as the argument to a method/li>
- Handle a checked exception in a Java application.

Development Tools
OS: MS Windows platform with JDK 8
IDE: NetBeans/Eclipse

METHODOLOGY
This program will be conducted with interactive lectures, PowerPoint presentation, discussions and practical exercise

COURSE OUTLINES

Module 1: *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

Module 2: *Creating a Java Main Class*
- Java Classes
- The main Method

Module 3: *Data In the Cart*
- Introducing variables
- Working with Strings
- Working with numbers
- Manipulating numeric data

Module 4: *Managing Multiple Items*
- Working with Conditions
- Working with a List of Items
- Processing a list of items

Module 5: *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
- Introducing the NetBeans IDE
- Introducing the Soccer League Use Case

Module 6: *Manipulating and Formatting the Data in Your Program*
- Using the String Class
- Using the Java API Docs
- Using the StringBuilder Class
- More about primitive data types
- The remaining numeric operators
- Promoting and casting variables

Module 7: *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

Module 8: *Using Encapsulation*
- Access Control
- Encapsulation
- Overloading constructors

Module 9: *More on Conditionals*
- Relational and conditional operators
- More ways to use if/else constructs
- Using Switch Statements
- Using the NetBeans Debugger

Module 10: *More on Arrays and Loops*
- Working with Dates
- Parsing the args Array
- Two-dimensional Arrays
- Alternate Looping Constructs
- Nesting Loops
- The ArrayList class

Module 11: *Using Inheritance*
- Overview of inheritance
- Working with subclasses and superclasses
- Overriding methods in the superclass
- Introducing polymorphism
- Creating and extending abstract classes

Module 12: *Using Interfaces*
- Polymorphism in the JDK foundation classes
- Using Interfaces
- Using the List Interface
- Introducing Lambda expressions

Module 13: *Handling Exceptions*
- Handling Exceptions: An overview
- Propagation of exceptions
- Catching and throwing exceptions
- Handling multiple exceptions and errors