Java8P1 - *Java SE 8 Programmer I*

DURATION: 5 Days; Instructor-led

WHAT YOU WILL LEARN
From developing apps to creating web applications to powering over 3 billion devices across the world, the Java programming language is ever present in today's world. When you become a student of Java – and eventually, an Oracle Certified expert – you create limitless opportunities for exciting jobs in the technology industry.

This course offers expert-led courses for beginner to advanced Java developers, covering core concepts, such as language constructs and data types, intermediate and advanced concepts, such as modular programming, secure coding, and convenience methods.

This entry-level course is aimed at programmers who are new to Java and who need to learn its concepts, language constructs, and data types. Included in the agenda are topics on exception handling, lambda expressions, and modular programming.

The course is designed for programmers who will apply these language skills to develop programs using the latest major versions of the JDK. Students practice the skills learned in each lesson through hands-on labs

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
- 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: *Java Basics*
- Define the scope of variables
- Define the structure of a Java class
- Create executable Java applications with a main method; run a Java program from the command line; produce console output
- Import other Java packages to make them accessible in your code
- Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation, etc.

Module 2: *Working with Java Data Types*
- Declare and initialize variables (including casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Know how to read or write to object fields
- Explain an object's lifecycle (creation, "dereference by reassignment" and garbage collection)

Module 3: *Using Operators and Decision Constructs*
- Use Java operators; use parentheses to override operator precedence
- Test equality between strings and other objects using == and equals()
- Create and use if, if-else, and ternary constructs
- Use a switch statement

Module 4: *Creating and Using Arrays*
- Declare, instantiate, initialize and use a one-dimensional array
- Declare, instantiate, initialize and use a multi-dimensional array

Module 5: *Using Loop Constructs*
- Create and use while loops
- Create and use for loops including the enhanced for loop
- Create and use do-while loops
- Compare loop constructs
- Use break and continue

Module 6: *Working with Methods and Encapsulation*

- Create methods with arguments and return values, including overloaded methods
- Apply the static keyword to methods and fields
- Create and overload constructors; differentiate between default and user defined constructors
- Apply access modifiers
- Apply encapsulation principles to a class
- Determine the effect upon object references and primitive values when they are passed into methods that change the values

Module 7: *Working with Inheritance*

- Describe inheritance and its benefits
- Develop code that makes use of polymorphism; develop code that overrides methods; differentiate between the type of a reference and the type of an object
- Determine when casting is necessary
- Use super and this to access objects and constructors
- Use abstract classes and interfaces

Module 8: *Handling Exceptions*

- Differentiate among checked exceptions, RuntimeException, and Error
- Create a try-catch block and determine how exceptions alter normal program flow
- Describe the advantages of exception handling
- Create and invoke a method that throws an exception
- Recognize common exception classes and categories (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)

Module 9: *Working with Selected classes from the Java API*

- Manipulate data using the StringBuilder class and its methods
- Create and manipulate strings
- Create and manipulate calendar data using classes from java.time.LocalDateTime, java.time.LocalDate, java.time.LocalTime, java.time.format.DateTimeFormatter, java.time.Period.
- Declare and use an ArrayList of a given type
- Write a simple Lambda expression that consumes a Lambda Predicate expression