

5 Days

Java SE 8 Programming Course

The Java SE 8 Programming training gives an overview of the core language features and Application Programming Interfaces (API) that can be used to design object-oriented applications with Java Standard Edition 8 (Java SE 8) Platform.

Completion of this training course will enable candidates in/to:

- Create high-performance multi-threaded applications
- Building Java technology applications with the latest JDK Technology
- Identifying good practices in the use of the language to create robust Java application
- Developing object-oriented skills
- Manipulating files, directories and file systems
- Storing and manipulating data deploying collections
- Connecting to databases deploying standard SQL queries through JDBC
- Deploying Lambda expressions in Java applications

Course Benefits:

Attaining certification of this course helps individuals to enhance their core skills with the Java language and also an essential prerequisite for the Oracle Certified Professional, Java SE 8 Programmer Exam.

Course Details

Course Outline

1. Java Platform Overview

- Define how the Java language achieves platform independence
- Differentiate between the Java ME, Java SE, and Java EE Platforms
- Evaluate Java libraries, middleware, and database options
- Define how the Java language continues to evolve

2. Java Syntax and Class Review

- Create simple Java classes
- Create primitive variables
- Use operators
- How to Use if-else and switch statements

- Create and manipulate strings
- Iterate with loops: while, do-while, for
- Use Java fields, constructors, and methods

3. Encapsulation and Sub classing

- Use encapsulation in Java class design
- Modeling business problems deploying Java classes
- Make classes immutable
- Create and use Java subclasses
- Overload methods

4. Overriding Methods, Polymorphism, and Static Classes

- Use access levels: private, protected, default, and public.
- Overriding methods
- Use virtual method invocation
- Use upward and downward casts
- Use var args to specify variable arguments
- Use the instanceof operator to compare object types
- Model business problems by deploying the static keyword
- Implement the singleton design pattern

5. Abstract and Nested Classes

- Designing general-purpose base classes by use abstract classes
- Constructing abstract Java classes and subclasses
- Apply final keyword in Java
- Distinguish between top-level and nested classes

6. Interfaces and Lambda Expressions

- Define a Java interface
- Choose between interface inheritance and class inheritance
- Extend an interface
- Default methods
- Anonymous inner classes
- Define a Lambda Expression

7. Collections and Generics

- Create a custom generic class
- Use the type inference diamond to create an object

- Create a collection by using generics
- Implement an ArrayList
- Implement a TreeSet
- Implement a HashMap
- Implement a Deque
- Order collections

8. Collections Streams, and Filters

- Describe the Builder pattern
- Iterating through a collection deploying lambda syntax
- Describe the Stream interface
- Filtering a collection deploying lambda expressions
- Calling an existing method deploying a method reference
- Chaining multiple methods together
- Define pipelines in terms of lambdas and collections

9. Lambda Built-in Functional Interfaces

- Core interfaces - Predicate, Consumer, Function, Supplier
- Listing the built-in interfaces included in java.util.function
- Use primitive versions of base interfaces
- Use binary versions of base interfaces

10. Lambda Operations

- Extracting data from an object using map
- Describe the types of stream operations
- Describe the Optional class
- Describe lazy processing
- Sorting a stream
- Saving results to a collection deploying the collect method
- Grouping and partition data deploying the Collectors class

11. Exceptions and Assertions

- Define the purpose of Java exceptions
- How to Use the try and throw statements
- How to Use the catch, multi-catch, and finally clauses
- Autoclose resources with a try-with-resources statement
- Explore common exception classes and categories

- Create custom exceptions
- Testing invariants by deploying assertions

12. Java Date/Time API

- Create and manage date and time based events
- Combine date and time into a single object
- Working with dates and times across time zones
- Manage changes resulting from daylight savings
- Define and create timestamps, periods and durations
- Applying formatting to local and zoned dates and times

13. I/O Fundamentals

- Describe the basics of input and output in Java
- Read and write data from the console
- Use streams to read and write files
- Writing and read objects deploying serialization

14. File I/O (NIO.2)

- Use Stream API with NIO2
- Use the Path interface to operate on file and directory paths
- Use the Files class to check, delete, copy, or move a file or directory

15. Concurrency

- Describe operating system task scheduling
- Create worker threads deploying Runnable and Callable
- Use an Executor Service to concurrently execute tasks
- Identify potential threading problems
- Use synchronized and concurrent atomic to manage atomicity
- Use monitor locks to control the order of thread execution
- Use the java.util.concurrent collections

16. The Fork-Join Framework

- Parallelism
- The need for Fork-Join
- Work stealing
- Recursive Task

17. Parallel Streams

- Review the key characteristics of streams
- Describe how to make a stream pipeline execute in parallel
- Describe the process for decomposing and then merging work
- Describe why reduction requires an associative function
- List the key assumptions needed to use a parallel pipeline
- Calculating a value deploying reduce
- List the key performance considerations for parallel streams

18. Database Applications with JDBC

- Define the layout of the JDBC API
- Connect to a database by deploying a JDBC driver
- Submit queries and get results from the database
- Specify JDBC driver information externally
- Perform CRUD operations deploying the JDBC API

19. Localization

- Read and set the locale by deploying the Locale object
- Define what a locale represents
- Describe the advantages of localizing an application
- Build a resource bundle for each locale
- Call a resource bundle from an application
- Change the locale for a resource bundle

Who Should Attend

The Java SE 8 Programming Certification course is ideal for:

- Developer
- Java Developers
- Java EE Developers

Pre Requisite

Understanding of Java SE 8 Fundamentals

464, Udyog Vihar Phase
V, Gurgaon (Delhi
NCR)-122016, India

+91 8882 233 777

training@mercury.co.in

www.mercurysolutions.co

Date - Apr 20, 2024