



Hewlett Packard Enterprise

Course Datasheet

Core Java

Education Services course product number – HPE-CJ-v1.0

Course length – 60 Hrs.

Delivery mode – Instructor Led Training (ILT)

Virtual Instructor Led Training (vILT)

Throughout the course, hands-on exercises (both classroom & Lab Assignment) are designed to teach object oriented programming using the Java Standard Edition programming language (JSE 8)

Course Objective

Learn the syntax, semantics and idioms of the Java programming language. Gain confidence in object-oriented programming principles through lots of practical exercises that provide useful exposure to the core Java class libraries.

Prerequisite

No experience is required. But fundamental knowledge of C/C++ would be helpful.

Course Modules

Chapter 01 – Introduction to JAVA Technology

- Introduction to JAVA Technology
- JAVA Language Features
- What is JAVA bytecode? What is JVM?
- Different Editions of JAVA
- Writing the “Hello World” command-line Application
- Understanding the “main()” method
- Setting path & classpath

Chapter 02 – Language Fundamentals I

- Keywords
- Primitive Data types
- Legal & Illegal Identifiers
- Different types of Literals
- String Literals
- Different types of JAVA Comments

Course Datasheet

- Declaring & Initializing variables
- Declaring constants using "final" keyword
- Object-oriented Features
- Implementing OO Concepts: Defining Classes
- Variables and methods as members of a class
- Object reference variables
- Restricting the access to object's members using an access specifier
- Different types of access specifiers
- Understanding the proper encapsulation

Chapter 03 – Language Fundamentals II

- Lifetime of Variables
- Instance methods and class methods
- Understanding Polymorphism
- Types of polymorphism
- Implementing static polymorphism in JAVA
- Understanding Inheritance
- Implementing single inheritance using "extends" keyword
- Implementing dynamic polymorphism
- Using "final" keyword for
- Abstract methods & Abstract classes

Chapter 04 – Language Fundamentals III

- Understanding the need of packages
- Creating a package
- Compiling and Running Code from Packages
- Importing a package
- Understanding the proper use of "protected" keyword
- Operation on Primitives: Using Operators
- Different types of operators
- Precedence & associativity rules for operators
- Evaluation order of operands
- Conversions
- Numeric Promotions:- Unary and Binary Numeric Promotions
- Floating-point arithmetic and "strictfp"

Chapter 05 – Language Enhancements (Java 7)

- Annotation
- Strings in Switch
- The try-with-resources Statement
- Type inference for generic instance creation
- Implementing binary literals
- Catching multiple exception types and rethrowing exceptions with improved type checking
- Improved varargs method invocations

Chapter 06 – Arrays, String and Wrapper Class

- Working with Arrays in JAVA
- Method Overloading Issues : using var-args methods
- Working with Strings
- Primitive Wrappers

Course Datasheet

Chapter 07 – Concept of interface, Abstract class and Exception Handling

- Interface
- What is Exception
- Exception class-hierarchy
- Understanding the difference between checked & unchecked exception
- Understanding stack-based execution
- Handling exceptions(using try-catch-finally blocks)
- Proper use of throw and throws clauses
- Chained Exception

Chapter 08 – Nested Class

- Overview of nested classes and interfaces
- Types of Nested Class
- Complete understanding of static nested class
- Complete understanding of non-static nested classes/Inner classes
- Local classes
- Anonymous classes
- Anonymous Interface

Chapter 09 – Multi-threading

- Introduction
- What is a Thread
- Asynchronous Behaviour/ Race condition
- Creating a thread
- Thread Life Cycle, states and their transition
- Using methods from java.lang.Thread for state transition
- Need of synchronization? Concept of Object Locking
- Language-level support for synchronization : “synchronized” keyword
- Defining synchronized methods & understanding its importance
- Using synchronized block
- Inter-thread communication
- Deadlock condition

Chapter 10 – Stream-Based I/O

- Understanding Streams
- Types: character and Binary streams
- Input and Output Streams
- Understanding Stream class Hierarchy
- Line-oriented I/O
- Buffered Streams
- Scanning and Formatting
- I/O from the command-line
- Standard streams
- The console
- Data streams
- Random Access File
- Serialization

Course Datasheet

Chapter 11 – NIO

- New APIs for file system access
- The java.nio.file package

Chapter 12 – Generics

- Understanding compile-time type-safety
- Introduction to Generics
- Defining a Generic class
- Understanding Type Parameters
- Understanding Type Erasure Process
- Sub-typing and Super-typing
- Using wildcards
- Using “? extends T” and “? Super T”
- Generic methods

Chapter 13 – Collection Framework

- Introduction to collections
- What is a collection framework
- Benefits of collection framework
- Interface Hierarchy

Chapter 14 – The java.util.Collection interface: Supported Operations

- Basic Operations
- Bulk Operations
- Array Operation

Chapter 15– Traversing collections

- Using for-each loop
- Using Iterator

Chapter 16 – The java.util.List interface: Supported Operations

- Positional Access
- Search
- Iteration
- Range-View

Chapter 17 – The java.util.Set interface: Supported Operations

- Basic Operations
- Bulk Operations
- Array Operations

Chapter 18 – Language Enhancements (Java 8)

- Lamda Expressions
- Generic Type changes and improvements
- Stream Collection Types
- Functional Interfaces

Course Datasheet

- Date/Time changes
- Type Annotations
- Other Changes