



## Java Basic Level

### About this course

This Basic level Java course introduces participants to the skills they need to successfully write their first application in Java. This course is optimized for Java SE 8/9 but earlier versions can be taught on request.

**Duration:** 5 days

**Participants:** Min 5, Max 15, Optimal 12

### Learning outcomes

Upon completion of this course participants will be able to:

- Apply OOP principles
- Design OO applications using abstraction, encapsulation, modularity and hierarchy
- Use Java exceptions to guarantee graceful error recovery for your application
- Create and use inner classes in your program to make them concise and clear
- Compile and execute programs with the JDK development tools and with an Integrated Development Environment (IDE) of your choice
- Use the core Java libraries (java.lang, java.util)
- Understand how to access a Database with JDBC and JPA

### Who should attend

This course is suitable for those wishing to learn Java for the first time. Some knowledge of Object Orientated Programme is an advantage.

### Course syllabus

#### **Module 1: Introduction to Java**

- Language and Platform Features
- The Java SE Development Kit (JDK)
- Working with the Development Environment

#### **Module 2: Classes and Objects**

- Object Model and Object-Oriented Programming
- Classes, References, and Instantiation
- Adding Data to a Class Definition, Adding Methods
- Encapsulation & Access Control
- Constructors and Initialization
- Static Members of a Class

- Scopes, Blocks, References to Objects

#### **Module 3: Flow of Control**

- Branching: if, if-else, switch
- Iteration: while, do-while, for, break, continue

#### **Module 4: Strings and Arrays**

- String, StringBuffer, StringBuilder
- Arrays and Varargs

#### **Module 5: Packages**

- Using Packages to Organize Code
- import statements
- Creating Packages, package Statement, Required Directory Structure
- Finding Classes, Packages and Classpath

#### **Module 6: Composition and inheritance**

- Using Composition to Deal With Complexity
- Composition/HAS-A, Delegation
- Using Inheritance and Polymorphism to share commonality
- IS-A, extends, Inheriting Features, Overriding Methods, Using Polymorphism
- Class Object and Abstract Classes

#### **Module 7: Interfaces**

- Using Interfaces to Define Types
- Interfaces and Abstract Classes
- Default Methods and static Methods
- Using Interfaces to Remove Implementation Dependencies

#### **Module 8; Exceptions**

- Exceptions and the Exception Hierarchy
- try and catch
- Handling Exceptions,
- Program Flow with Exceptions

#### **Module 9: Java Collections and Generics**

- The Collections Framework and its API
- Collections and Java Generics
- Collection, Set, List, Map, Iterator, Autoboxing
- Collections of Object (non-generic)
- Using ArrayList, HashSet, and HashMap
- Processing Items With an Iterator

#### **Module 10: Database Access with JDBC and JPA**

- JDBC Overview
- JDBC Architecture and API



**CENIT**  
GROUP  
*study learn grow*

**Module 11: Additional Java Features**

- Assertions
- Type-safe Enums
- Annotations
- Lambda Expressions and Method References
- Additional Features

**Module 12: I/O Streams (Optional)**

- Readers and Writers
- Filter Streams
- Byte Streams
- Formatted Output
- New I/O (NIO) APIs