

The Java 12 Language

OO, Classes, Inheritance, Interfaces, Generics, Annotations, Exceptions, Events, Modules, JNI

With a much improved standards update cadence ([Java 12 was release in March 2019](#)) there is re-invigorated interest in advancing the Java language and ecosystem, by Oracle and external teams using and contributing to Java's evolution.

The Java 12 Language is a powerful object-oriented language with modern features such as the Java Platform Module System, lambdas, generics and annotations and a comprehensive set of OO features such as classes, inheritance, interfaces and exceptions.

Some of the largest enterprise projects in the world are written in Java. These are multi-year projects that are vital corporate assets and need to continuously evolve well into the future. Java is an important language for many university courses. Many high throughput cloud platforms are written in Java. The main language for Android is Java. Plenty of cutting-edge open source projects are written in Java. In addition to existing projects, Java is also being regularly selected as the main language for all kinds of new projects. Hence now is an ideal time to learn Java well – starting with the language itself.

Contents of One-Day Training Course	
<p>Target Audience Software developers wishing to become Java developers, starting with learning the language itself.</p> <p>This is an ideal first course in Java.</p> <p>Prerequisites Programming experience with an OO language such as C++, TypeScript or C#, along with good exposure to object-oriented design.</p> <p>No previous experience of Java is needed, as this course covers the language from the fundamentals up.</p>	<p>Java - Language & Ecosystem Review of the ecosystem surrounding the Java language and how it is evolving Features - language, VM, bytecode, runtime services, framework, tooling What's new in Java 12</p> <p>A Java Project Walkthrough Simple hello world project Main and command line args Primitive data types Code layout & basic syntax Classpath / JAR files</p> <p>Classes And Constructors Simple class definitions Static vs. instance members Access modifiers Variety of constructor layouts Nested classes</p> <p>Inheritance Inheritance trees Extending classes Overriding / hiding</p> <p>Generics Generic functions Different invocations Generic types (type variables) Shadowing Constraints</p> <p>Annotations We often wish to attach additional data to a class -without impacting the inheritance hierarchy Java annotations allow use of metadata</p> <p>Interfaces & Mixins Defining behaviors via interfaces Implementing multiple interfaces Uses for interfaces Default methods How to provision mixins using interfaces</p> <p>Exceptions Basic exception handling Creating an exception Throwing and re-throwing Catching an exception</p> <p>Event Handling java.lang.EventObject Observer and Observable Creating event listeners</p> <p>Lambdas Lambda expressions Quick and easy way to define a method Arguments and body Designing code using lambdas</p> <p>Java Platform Module System A named set of packages Module keyword Organizing modules</p> <p>JNI - Calling C Code JNI – Java Native Interface Calling out to C code from Java Passing parameters and accepting returning results between different languages</p> <p>Java Project Developing a Java project using a selection of language features showing how they can sensibly be used together</p>