

Why Java?

- general purpose programming .
- It has designed for easy developing run on desktops, servers, and mobile devices
- Java is a consolidation language, designed to absorb the strengths of earlier languages and ignore their weakness

Java advantages

o Simple, Object-Oriented, Distributed, Interpreted, Robust, Secure, Architecture-Neutral, Portable, High-Performance, Multithreaded, Dynamic

Portable

- With a Java Virtual Machine (JVM), you can write one program that will run on any platform.

Types of Java Applications:

Standalone Applications: Run on local computer (J2SE)

Applet: Run under a Java-enabled Web browser (J2SE))

Midlet: Run on a Java-Enabled Mobile Phone (J2ME)

Servlet: Run on a Java-enabled Web server (J2EE):

Java Development Kit (JDK):

tools that will be required to develop Java applications.

JDK = JRE + Required tools to develop Java Application

Java Runtime Environment (JRE):

Contains everything required to run Java applications. It contains JVM + JIT compiler, class libraries, and other supporting files

JRE = JVM + Required Library to run Java Application

Java Virtual Machine (JVM):

Works on top of your operating system to provide a platform-independent way of executing Java bytecode.

Execution Model of Java Programs

Phase 1: editing a file with an editor

- o Type a Java program (source code) using the editor , Make any necessary corrections , Save the program
- o A file name ending with the .java extension indicates that the file contains Java source code

Phase 2: compiling a Java program into bytecodes

- o Use the Java compiler (javac) to compile a Java program
- o If the program compiles, the compiler will produce a .class file known as bytecode

Phase 3: Loading a Program into Memory

- o The JVM places the program in memory to execute it — loading

- o Class loader takes the .class files containing the program's bytecodes and transfers them to primary memory

Phase 4: Bytecode Verification

- o As the classes are loaded, the bytecode verifier checks their bytecodes to ensure that they're valid and don't violate Java's security restrictions

Phase 5: Execution

- o The JVM executes the program's bytecodes using a combination of interpretation and so-called just-in-time (JIT) compilation
- o A just-in-time (JIT) compiler—known as the Java HotSpot compiler translates the bytecodes into machine-language code at runtime

Anatomy of a Java Program

Comments, Reserved words, Modifiers, Identifiers, Statements, Blocks, Classes, Methods, The main method

Comments should be included to explain the purpose of the program and describe processing steps

- o Line comment: `// this comment runs to the end of the line`
- o Paragraph comment: `/*this comment runs to the terminating symbol, even across line breaks */`
- o javadoc comment: `/** this is a javadoc comment*/`

The Java reserved words:

Have a specific meaning to the compiler and cannot be used for other purposes in the program

modifiers:

- o **Access modifiers** – specify accessibility of a data member, method, constructor, or class and how they can be used: private, public, protected, and default
- o **Non-Access modifiers** – provide functionalities other than the visibility

The static modifier for creating class methods and variables

The final modifier for finalizing the implementations of classes, methods, and variables

The abstract modifier for creating abstract classes and methods

The synchronized and volatile modifiers are used for threads

The transient modifier is used in serialization process

Identifiers are the words a programmer uses in a program , consists of alphabetical characters and digits (plus the underscore '_' and the dollar sign '\$' characters)

start with an alphabetic character, underscore, dollar sign but cannot start with a digit

No spaces are allowed inside the identifier's name, case sensitive - Total, total, and TOTAL are different identifiers , Not reserved words

Java Statements

Is a command for the computer to do something. Any statement in Java should be ended by a semicolon (;)

Blocks: A pair of curly braces in a program forms a block that groups components of a program

Classes: The class is the essential Java construct. It is a template (blueprint) for objects.

A method is a set of code which is referred to by name and can be called at any point in a program. Think of it as a subprogram that acts on data and often returns a value.

The main method provides the control of program flow

The Java interpreter executes the application by invoking the main method

Errors

A program can have three types of errors:

- o **Syntax (compile-time) error:** A grammatical mistake in a program

- ↳ If compile-time errors exist, an executable version of the program is not created

- o **Run-time error:** A problem can occur during program execution, such as trying to divide by zero, which causes a program to terminate abnormally

- o **Logic error:** A program may run, but produce incorrect results, perhaps using an incorrect formula