Java SE 8 Fundamentals

INTRODUCTION

Course Objectives

After completing this course, you should be able to:

- Demonstrate knowledge of basic programming language concepts
- Demonstrate knowledge of the Java programming language
- Implement intermediate Java programming and object-oriented (OO) concepts

Schedule

Day One

- Lesson 1: Introduction
- Lesson 2: What Is a Java Program?
- Lesson 3: Creating a Java Main Class
- Lesson 4: Introducing variables

Schedule

Day Two

- Lesson 5: Managing Multiple Items
- Lesson 6: Describing Objects and Classes

Day Three

- Lesson 7: Manipulating and Formatting the Data in Your Program
- Lesson 8: Creating and Using Methods

Schedule

Day Four

- Lesson 9: Using Encapsulation
- Lesson 10: More on Conditionals

Day Five

- Lesson 11: Working with Arrays, Loops, and Dates
- Lesson 12: Using Inheritance
- Lesson 13: Using Interfaces

How Do You Learn More After the Course?

To find more resources, bookmark the URL:
Oracle.com/oll/java
Look for the *Java SE 8 Fundamentals Collection*.

What is Java Program

Objectives

After completing this lesson, you should be able to:

- Contrast the terms “platform-dependent” and “platform-independent”
- Describe the purpose of the JVM
- Explain the difference between a procedural program and an object-oriented program
- Describe the purpose of `javac` and `java` executables
- Verify the Java version on your system
- Run a Java program from the command line

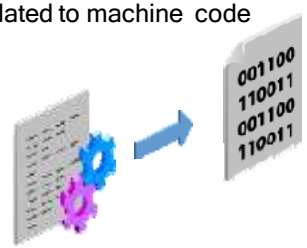
Topics

- Introduction to computer programs
- Introduction to the Java language
- Verifying the Java development environments
- Running and testing a Java program

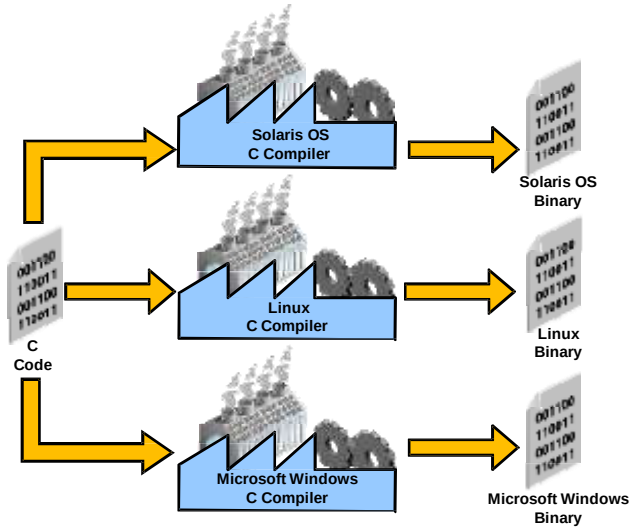
Purpose of a Computer Program

A computer program is a set of instructions that run on a computer or other digital device.

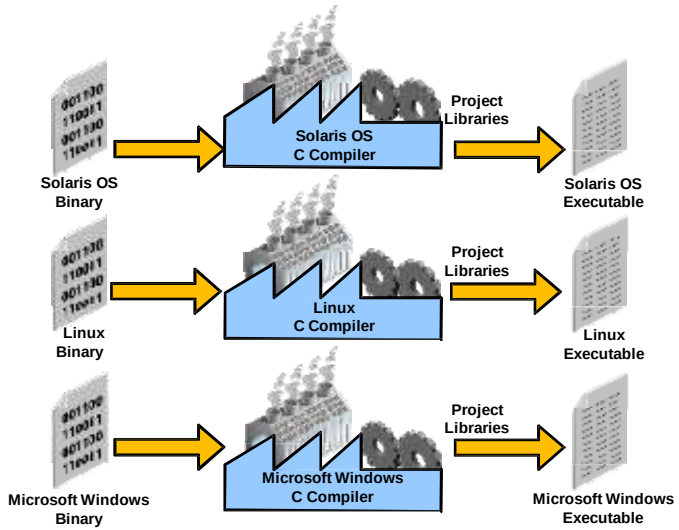
- At the machine level, the program consists of binary instructions (1s and 0s).
  - Machine code
- Most programs are written in *high-level* code (readable).
  - Must be translated to machine code



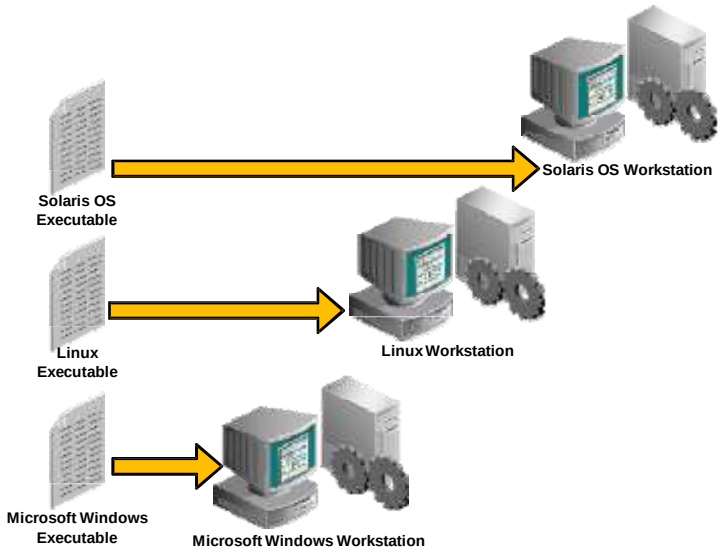
Translating High-Level Code to Machine Code



Linked to Platform-Specific Libraries



Platform-Dependent Programs



Topics

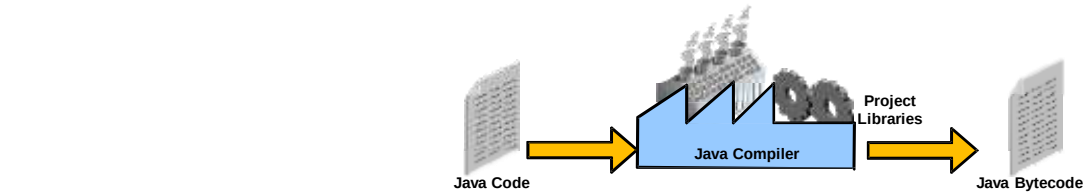
- Introduction to computer programs
- Introduction to the Java language
- Verifying the Java development environment
- Running and testing a Java program

Key Features of the Java Language

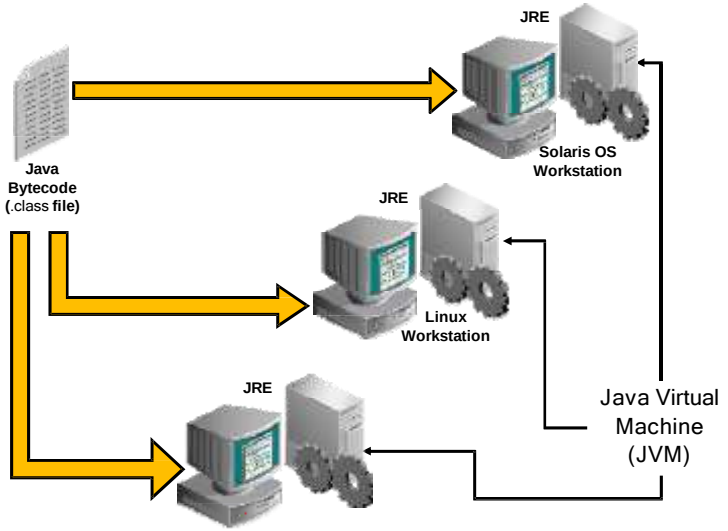
Some of the features that set Java apart from most other languages are that:

- It is platform-independent
- It is object-oriented

Java Is Platform-Independent

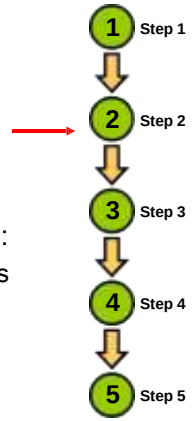


Java Programs Run In a Java Virtual Machine



Procedural Programming Languages

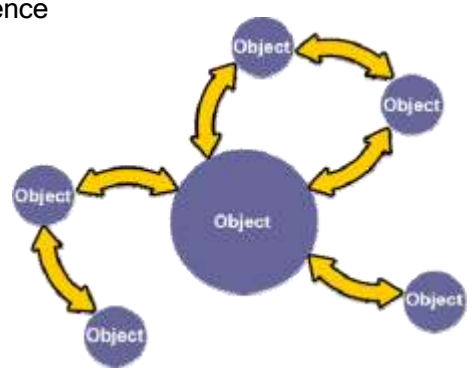
- Many early programming languages followed a paradigm called *Procedural Programming.*
- These languages use a sequential pattern of program execution.



- Drawbacks to procedural programming:
  - Difficult to translate real-world use cases to a sequential pattern
  - Difficult to maintain programs
  - Difficult to enhance as needed

Java Is an Object-Oriented Language

- Interaction of objects
- No prescribed sequence
- Benefits:
  - Modularity
  - Information hiding
  - Code reuse
  - Maintainability



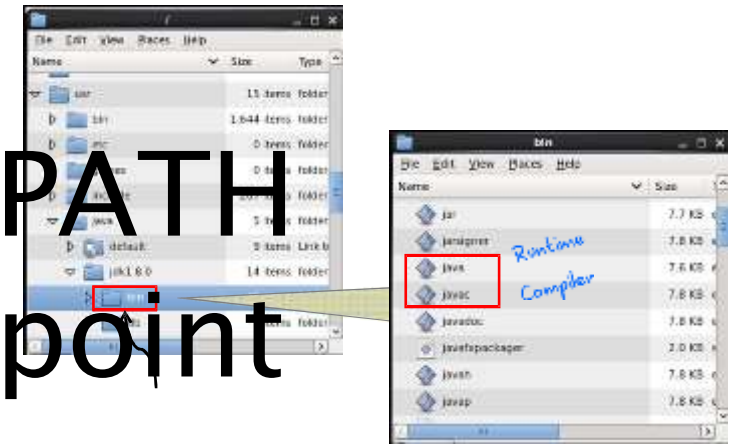
Topics

- Introduction to computer programs
- Introduction to the Java language
- Verifying the Java development environment
- Running and testing a Java program

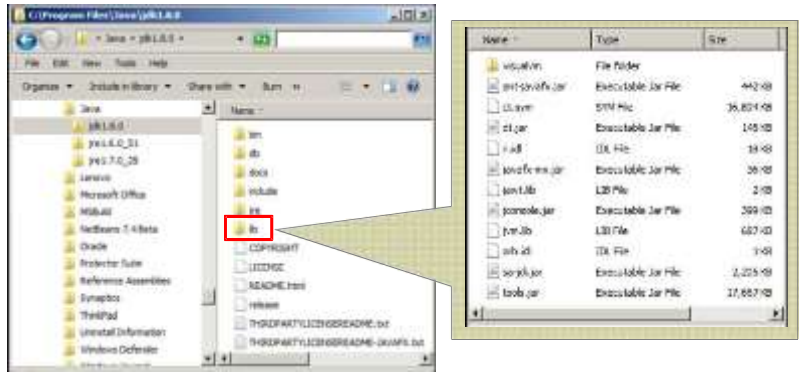
Verifying the Java Development Environment

1. Download and install the Java Development Kit (JDK) from oracle.com/java.
2. Explore the Java Help menu.
3. Compile and run a Java application by using the command line.

Examining the Installed JDK (Linux Example): The Tools



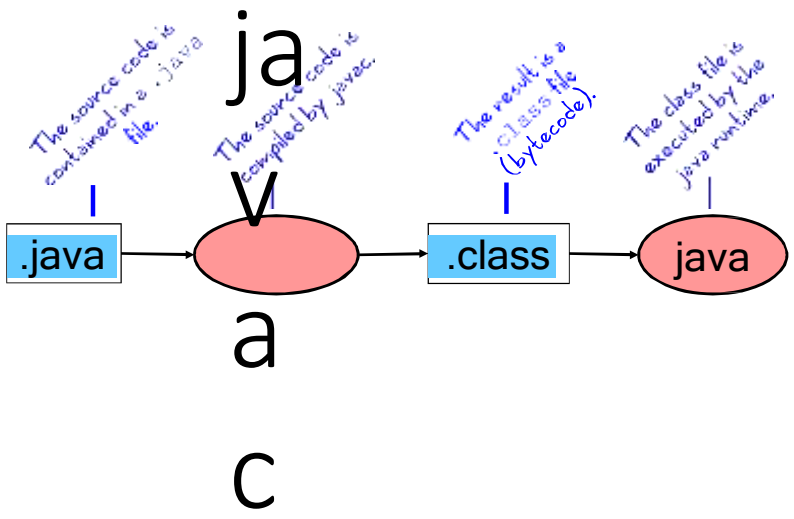
Examining the Installed JDK (Windows Example): The Libraries



Topics

- Introduction to computer programs
- Introduction to the Java language
- Verifying the Java development environment
- Running and testing a Java program

Compiling and Running a Java Program



Compiling a Program

1. Go to the directory where the source code files are stored.
2. Enter the following command for each `.java` file you want to compile.

- Syntax:

```
javac SayHello.java
```

- Example:

```
javac SayHello.java
```

Executing (Testing) a Program

1. Go to the directory where the class files are stored.
2. Enter the following for the class file that contains the `main` method:

- Syntax:

```
java <classname>
```

- Example:

Do *not* specify `.class`.

```
java SayHello
```

- Output:

```
Hello World!
```