



Unit objectives

- After completing this unit, you should be able to:
 - Describe the history and properties of the Java programming language
 - Explain the Java execution model, including the use of bytecode and the Java virtual machine
 - Outline the types of programs and components that can be built using Java

What is Java?

- Java is an object-oriented programming language developed by Sun Microsystems
- Java has a set of standardized class libraries that support predefined reusable functionality
- Java has a runtime environment that can be embedded in Web browsers and operating systems

Goals of Java

- Java was developed to solve several problems with existing languages
- These issues had an important influence on its development:
 - Object-orientation
 - Portability
 - Performance
 - Security
 - Internationalization
 - Distributed computing
 - Dynamic systems

The Java programming language(1/4)

■ Object-oriented

- Java supports software development using the notion of objects
- Software developed using Java is composed of classes and objects

■ Network capable

- Java supports the development of distributed applications
- Some types of Java applications are designed to be accessed through a Web browser

The Java programming language(2/4)

■ Robust

- Many aspects of Java promote the development of reliable software
- Java uses a pointer model which does not allow direct access to memory; memory cannot be overwritten

■ Secure

- Java authentication is based on public-key encryption methods
- Java's pointer model protects private data in objects and prevents unauthorized applications from accessing data structures

The Java programming language(3/4)

■ Multi-threaded

- Allows your program to run more than one task at the same time

■ Compiled and interpreted

- Source code is compiled into machine code for the Java virtual machine (JVM) by the Java compiler
- Machine code for the JVM is also known as bytecode
- Interpreter of the Java virtual machine interprets and executes instructions

The Java programming language(4/4)

■ Architecture neutral

- Bytecode instructions are architecture neutral because they run on the JVM, and are not specific to an architecture
- The same application runs on all platforms, provided the Java virtual machine is installed on that platform

■ Portable at source and binary level

- One piece of source code gets compiled into one set of bytecode instructions for the JVM, and can be run on any platform and architecture without recompiling the code

Java evolution(1/2)

- Java is a relatively young language
 - It has been in use since 1995
 - It was originally designed for consumer electronic devices(TV, VCR, Freeze, Washing Machine, Mobile Phone)
 - Internet and Web was just emerging, so Sun turned it into a language of Internet Programming.
 - It allows you to publish a webpage with Java code in it.
- Java has a huge developer base
 - There is a vast collection of libraries (from Sun and other sources)

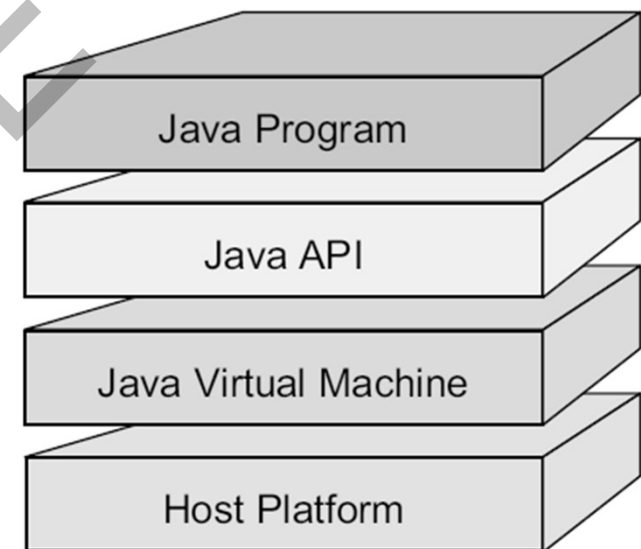
Java evolution(2/2)

■ Versions

- JDK 1.0 (January 23, 1996)
- JDK 1.1 (February 19, 1997)
- J2SE 1.2 (December 8, 1998)
- J2SE 1.3 (May 8, 2000)
- J2SE 1.4 (February 6, 2002)
- J2SE 5.0 (September 30, 2004)
- Java SE 6 (December 11, 2006)
- Java SE 7 (July 28, 2011)
- Java SE 8 (March 18, 2014)
- Java SE 9 (September 22, 2017)
- Java SE 10 (March 21, 2018)
- Java SE 11 (September 25, 2018)
- Java SE 12 (March 20, 2019)

The Java Platform

- A platform is a development or deployment environment
- The Java platform runs on any operating system
 - Other platforms are hardware and vendor specific
- The Java platform provides:
 - The Java virtual machine(JVM)
 - Application Programming Interface (API)



The Java execution model

- Compiled language execution model
 - Source code is compiled into instructions understood by the host machine; host machine executes this binary file



- Java execution model
 - Source code is compiled into instructions understood by the JVM; bytecode is then interpreted by the JVM



Use of Java

- Java can be used to build programs and software components
- Programs are stand-alone entities that can run on the Java virtual machine
 - Applications
 - Applets
- Components are building blocks used to create programs
 - Servlets
 - JavaServer Pages (JSPs)
 - JavaBeans
 - Enterprise JavaBeans (EJBs)

■ Programs

■ Application

- A stand-alone program that can access system resources such as files
- Does not need to run in a Web browser
- Is explicitly invoked through the command line or menu selection
- The method `main()` is the entry point for an application

■ Applet

- A Java program that is embedded within a Web page; almost always graphical
- Security limits access to system resources
- Code executes on the client inside a Web browser



Components(1/2)

■ Servlet

- Handles requests from the Web browser and returns responses
- Creates dynamic content on the server
- Runs inside an application server

■ JavaServer Page (JSP)

- HTML page embedded with Java code
- Creates dynamic content on the server instead of on the browser
- Runs inside an application server



Components(2/2)

■ JavaBeans

- Java code that has its properties, methods, and events exposed to promote reuse among developers
- Reusable software component that can be manipulated visually in a builder tool

■ Enterprise JavaBeans (EJB)

- Distributed objects that allow communication between Java objects in different JVMs
- Encapsulate the business logic and model of an application
- Run inside an application server



Demo

- A Simple Java Program
- (GUI) Displaying Text in a Message Dialog Box

XJTUSEE



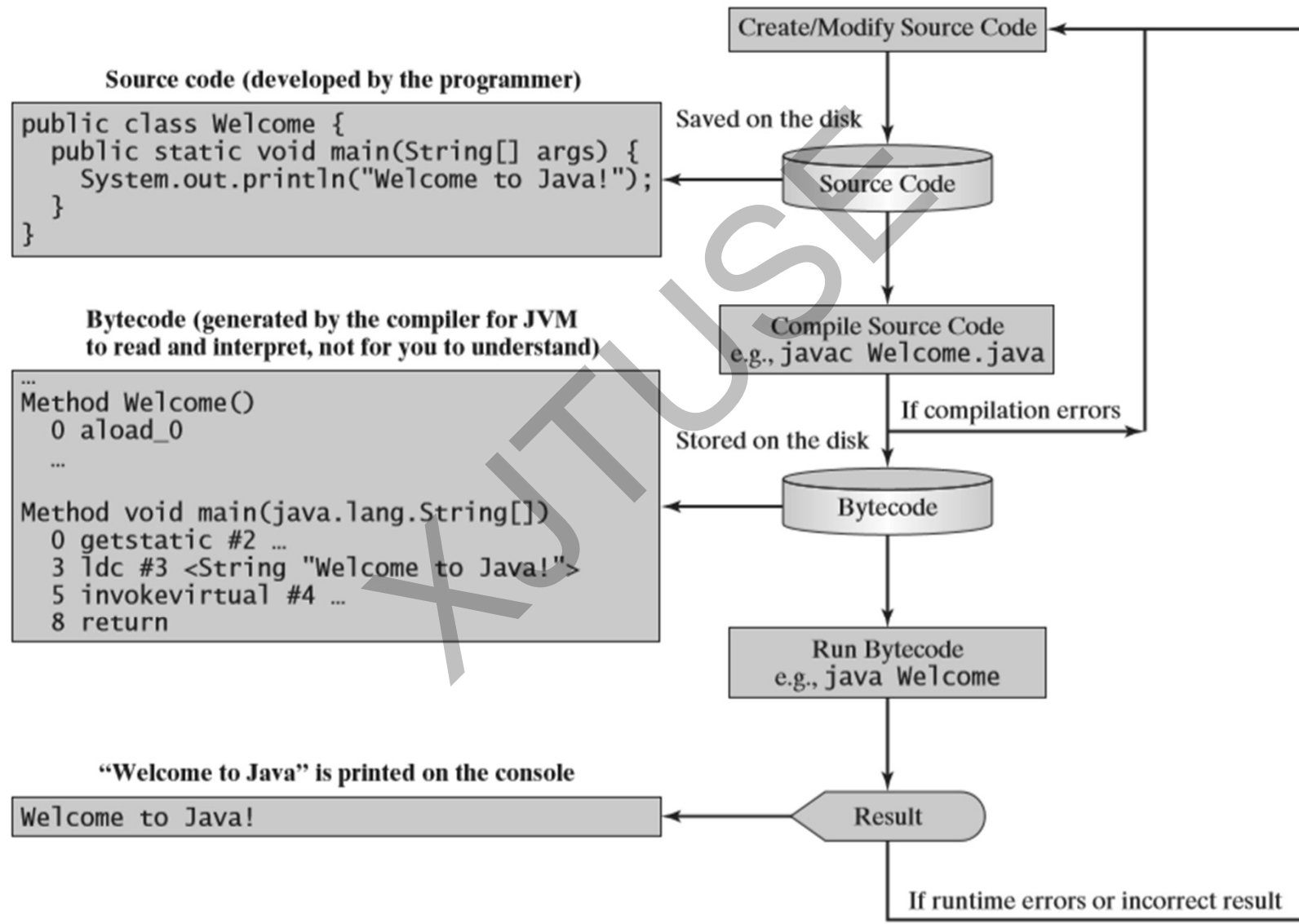
A Simple Java Program

LISTING 1.1 Welcome.java

```
1 public class Welcome {  
2     public static void main(String[] args) {  
3         // Display message Welcome to Java! to the console  
4         System.out.println("Welcome to Java!");  
5     }  
6 }
```

```
D:\>java Welcome  
Welcome to Java!
```

A Simple Java Program



(GUI) Displaying Text in a Message Dialog Box

LISTING 1.4 WelcomeInMessageDialogBox.java

```
1 /* This application program displays Welcome to Java!
2  * in a message dialog box.
3  */
4 import javax.swing.JOptionPane;
5
6 public class WelcomeInMessageDialogBox {
7     public static void main(String[] args) {
8         // Display Welcome to Java! in a message dialog box
9         JOptionPane.showMessageDialog(null, "Welcome to Java!");
10    }
11 }
```



Checkpoint

- 1. What is the JVM and what does it do?
- 2. What is bytecode?
- 3. What is the difference between a program and a component?
- 4. What are the differences between an application, an applet and a servlet?

???

Unit summary

- In this module, you should have learned to:
 - Describe the history and properties of the Java programming language
 - Explain the Java execution model, including the use of bytecode and the Java virtual machine
 - Outline the types of programs and components that can be built using Java