

1. Introduction

Java

Fall 2009

Instructor: Dr. Masoud Yaghini

Outline

- Introduction
- The Java Programming Language
- The Java Platform
- References



Introduction



Java technology

- **Java** is
 - A **high-level programming language**, and
 - A **platform**
- Originally implemented by James Gosling, first publicly released in 1995
- Applications:
 - Developing desktop, web, enterprise, and mobile applications

Advantages of Java (I)

- **Get started quickly**
 - It's easy to learn, especially for programmers already familiar with C or C++.
- **Write less code**
 - A program written in the Java programming language can be four times smaller than the same program written in C++.
- **Write better code**
 - Java let you reuse existing, tested code and introduce fewer bugs.

Advantages of Java (II)

- **Develop programs more quickly**
 - The Java programming language is simpler than C++, and as such, your development time could be up to twice as fast when writing in it.
- **Write once, run anywhere**
 - Because applications written in the Java programming language are compiled into machine-independent, they run consistently on any Java platform.

The Java Programming Language

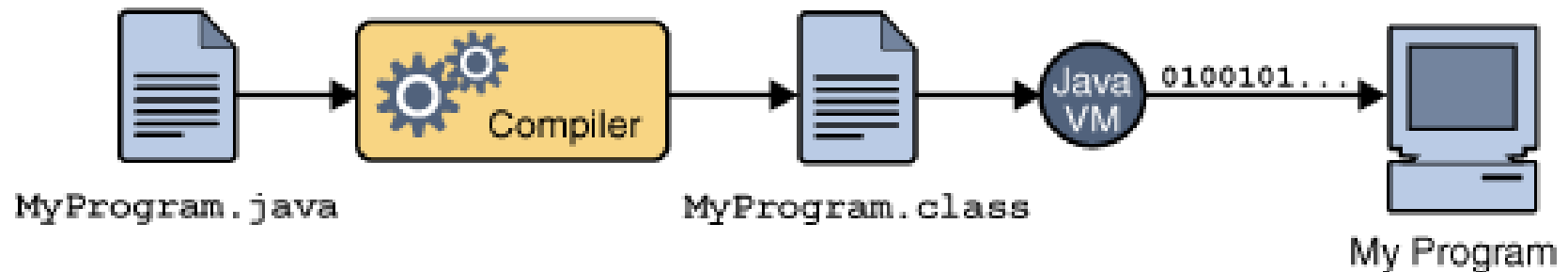


The Java Programming Language

- Java is a high-level programming language
- **Source code**
 - All **source code** is first written in text files ending with the **.java** extension.
- **Class files**
 - The source files are compiled into **.class** files by the **javac compiler**.
 - A **.class** file contains **bytecodes**
 - The **java launcher tool** then runs your application with an instance of the **Java Virtual Machine (Java VM)**.

The Java Programming Language

- An overview of the software development process:



The Java Programming Language

- Because the Java VM is available on many different platforms (operating systems), the same .class files are capable of running on different operating systems:
 - Microsoft Windows
 - Solaris
 - Linux
 - Mac OS

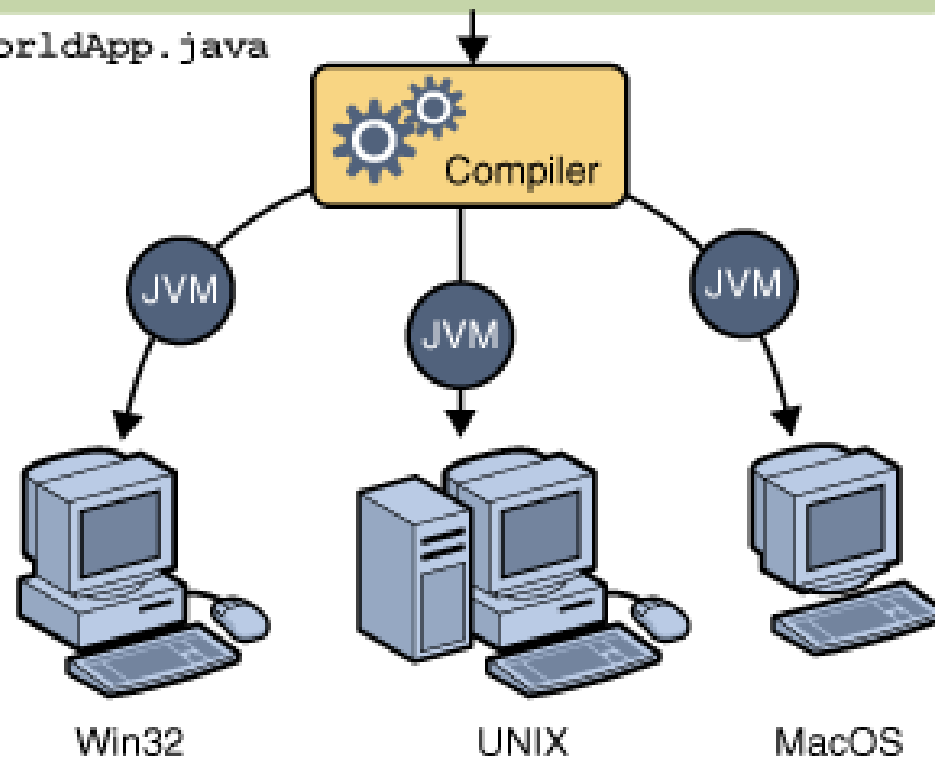
Introduction

The Java Programming Language

Java Program

```
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!");  
    }  
}
```

HelloWorldApp.java



Java Building Blocks

- **Classes**
 - Include **methods** that perform tasks
 - Methods return information after task completion
- When programming in Java, you will typically use the following building blocks:
 - Classes and methods from class **libraries**
 - Classes and methods that others create and make available to you
 - Classes and methods you create yourself



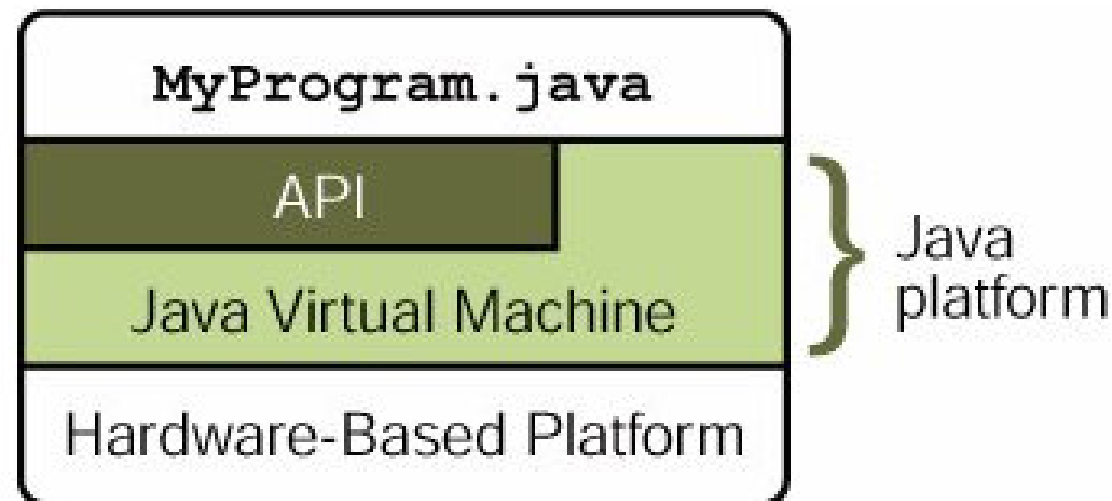
The Java Platform

The Java Platform

- **Platform**
 - **A platform** is the hardware or software environment in which a program runs.
 - Most platforms can be described as a combination of the operating system and underlying hardware.
- **Java platform**
 - **The Java platform** is a software-only platform that runs on top of other hardware-based platforms.

The Java Platform

- The Java platform components:
 - The **Java Virtual Machine**
 - The **Java Application Programming Interface (API)**



The Java Platform

- **Java Virtual Machine (VM)**
 - The Java Virtual Machine is the base for the Java platform and is ported onto various hardware-based platforms.
- **Java Application Programming Interface (API)**
 - The API is a large collection of ready-made software components that provide many useful capabilities.
 - API is grouped into libraries of related classes and interfaces; these libraries are known as **packages**.

Java Platforms

- There are three main platforms for Java:
 - **Java SE (Standard Edition)**
 - runs on desktops and laptops
 - **Java ME (Micro Edition)**
 - runs on mobile devices such as cell phones
 - **Java EE (Enterprise Edition)**
 - runs on servers
- This course uses Java SE to introduce Java programming.

Java Development Kit (JDK)

- **The Java Development Kit (JDK)**
 - **JDK** is a development environment for building applications, applets, and components using the Java programming language.
 - The JDK includes tools useful for **developing** and **testing** programs written in the Java programming language and running on the Java platform.

Java Runtime Environment (JRE)

- **The Java Runtime Environment (JRE)**
 - **JRE** allows you to **run** applications written in the Java programming language.
 - Like the JDK, it contains the **Java Virtual Machine (JVM)**, classes comprising the Java platform API, and supporting files.
 - Unlike the JDK, it does not contain development tools such as compilers and debuggers.
 - You can freely redistribute the JRE with your application, according to the terms of the JRE license.

JDK Versions

- JDK 1.02 (1995)
- JDK 1.1 (1996)
- JDK 1.2 (1998)
- JDK 1.3 (2000)
- JDK 1.4 (2002)
- JDK 1.5 (2004)
- JDK 1.6 (2006) a. k. a. JDK 6 or Java 6



Java IDEs



Java IDEs

- **Integrated Development Environments (IDE)** provide benefits to programmers.
- IDE allows advanced features like code generators, auto-completion, and debuggers.

Introduction

Some Popular Java IDEs

Java IDE	Free/Commercial	Company
Netbeans	Free	Sun Microsystems
Eclipse	Free	Open source
Sun Java Studio	Commercial	Sun Microsystems
JBuilder	Commercial	Borland
IntelliJ IDEA	Commercial	JetBrains
JDeveloper	Commercial	Oracle
IBM WebSphere Studio	Commercial	IBM



References



References

- S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, **The Java Tutorial: A Short Course on the Basics**, 4th Edition, Prentice Hall, 2006. (Chapter 1)



The End