CSC 3215: Object Oriented Programming - 1 (JAVA)

1. Introduction to Programming in Java

Dr. Kamruddin Nur

Associate Professor
Dept. of Computer Science, AIUB
kamruddin@aiub.edu

September, 2018

Contents

1 Textbook & Reference Books

Textbook & Reference Books

Textbook:

1. **Java the Complete Reference**, 9th Edition, by Herbert Schildt

Reference books:

1. **Java How to Program Java**, 9th Edition, By Deitel and Deitel
2. **Introduction to Programming in Java**, by Robert Sedgewick & Kevin Wayne
3. **Thinking in Java**, 4th Edition, by Bruce Eckel
4. **The Java Language Specification**, By J. Gosling, B. Joy, G. Steele, G.Bracha and A. Buckley
5. **The Java Tutorials**, docs.oracle.com/javase/tutorial

2 Course Instructor

Faculty Member/s

Dr. Kamruddin Nur

Associate Professor, Computer Science, AIUB

Office: **Annex 4, 42007**

Office Hours:
Sun 11 am - 12 pm, 4.30 pm - 6 pm
Mon 11 am - 1.30 pm, 4.30 pm - 6 pm
Wed 11 am - 1.30 pm, 5 pm - 6 pm

Contact: kamruddin@aiub.edu

3 Evaluation

Evaluation

Marking System for Mid Term		Marking System for Final Term	
Attendance & Performance	10	Attendance & Performance	10
Quiz (Best 1 out of 2)	20	Quiz (Best 1 out of 2)	20
Lab Tasks	20	Lab Tasks	20
Lab Exam	10	Project Viva	10
Mid Term Exam	40	Project	40
Total	**100**	**Total**	**100**
Grand Total: 50% of Mid Term + 50% of Final Term			

4 What is Java?
- Java Platforms
- JVM, JRE, and JDK
- Java Virtual Machine (JVM)

What is Java?

Java:

- Java is an object-oriented programming language designed by James Gosling developed at Sun Microsystems in the mid 1990s
- Originally called Oak then renamed to Java
- Syntax is very similar to C
- A programming language and a programming platform
- Major Releases: **Java 1.0** (1995), **Java 2.0** (1999), Free **Open-Source JVM** (2006)
- Major Feature: "Write Once, Run Anywhere" (**WORA**)
- Sun describes it as -

"A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multi-threaded and dynamic language."

Figure 1: James Gosling, the creator of Java

What is Java? *(Cont'd)*

- **Object-Oriented**
  - **Designed to support Object-Oriented concepts**
  - However, does contain non-Object-Oriented primitive data types (e.g. int, short, long, float, double, boolean, char etc.)
- **Distributed**
  - Applications are constructed using objects. **Objects can be distributed in multiple locations within a network environment**
  - Extensive integration with TCP/IP
- **Interpreted**
  - Java compiler compiles **source code to byte-code** (not machine code). **Byte code is interpreted**.
  - **JIT** (Just-In-Time) compiler compiles **byte code to machine code**

What is Java? *(Cont'd)*

- **Robust**
  - **Memory management is done automatically**
  - Use of pointers is limited
- **Secure**
  - **All Java code are subject to security model**
- **Architecture-Neutral / Portable**
  - Compiled Java (byte code) will **run on any platform which has a Java Virtual Machine**
  - The Java Virtual Machine is available for almost all platforms (i.e. Windows, Linux, Mac etc.)
- **High-Performance**
  - Java was always **intended to be high-performance**
  - Previously, Java's performance was poor, however, now, Java's performance resembles the rival C++.

What is Java? *(Cont'd)*

- **Multi-Threaded**
  - **A process can have multiple threads of execution**
  - Similar to multi-tasking but all threads share the same memory space
- **Dynamic**
  - **Makes heavy use of dynamic memory allocation**
  - Classes can be dynamically loaded at any time

Java Platforms

1. **J2SE** (Java 2 Standard Edition)
   - For developing standalone applications (softwares)
2. **J2EE** (Java 2 Enterprise Edition)
   - For developing enterprise web application
3. **J2ME** (Java 2 Micro Edition)
   - For developing applications that run on devices, mobiles
4. **JavaFX** (Java 2 + Flash + Flex)
   - For developing effect rich applications

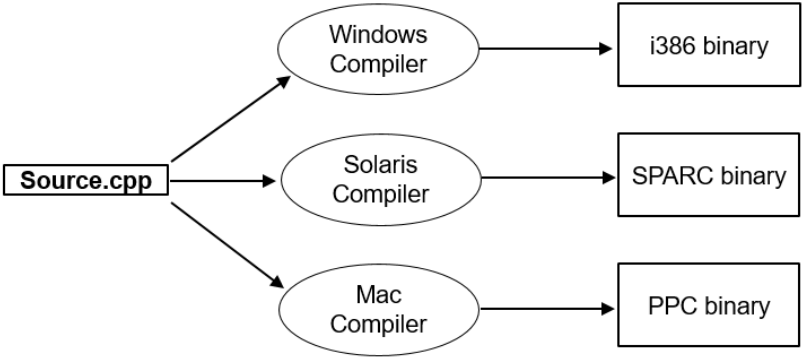
JVM, JRE, and JDK

- **JVM** (Java Virtual Machine)
  - Abstract machine
  - **Specification** that provides runtime environment in which java bytecode (generated by java compiler) can be executed
- **JRE** (Java Run-time Environment)
  - Implementation of JVM
  - Contains **JVM + Libraries**
  - Required to run a java program
- **JDK** (Java Development Kit)
  - Contains **JRE + Development tools** (i.e. **compilers and debuggers** for developing a java applications)
  - Required to develop and run a java program

Important: JVM, JRE and JDK are platform dependent, because configuration of each OS differs. However, Java is platform independent.

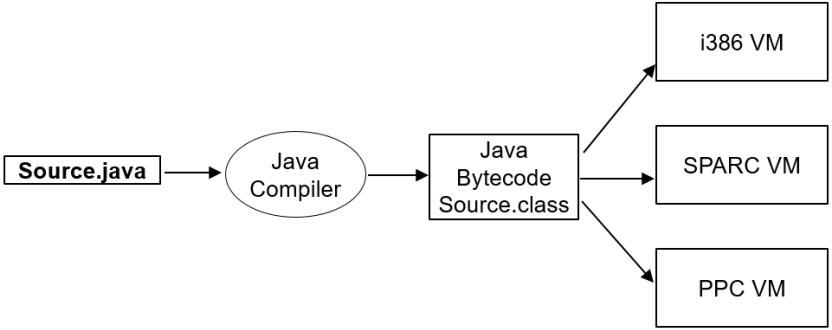
Java Virtual Machine (JVM)

- **Traditionally**, source code had to be compiled for the target hardware and OS platform:



Java Virtual Machine (JVM) *(Cont'd)*

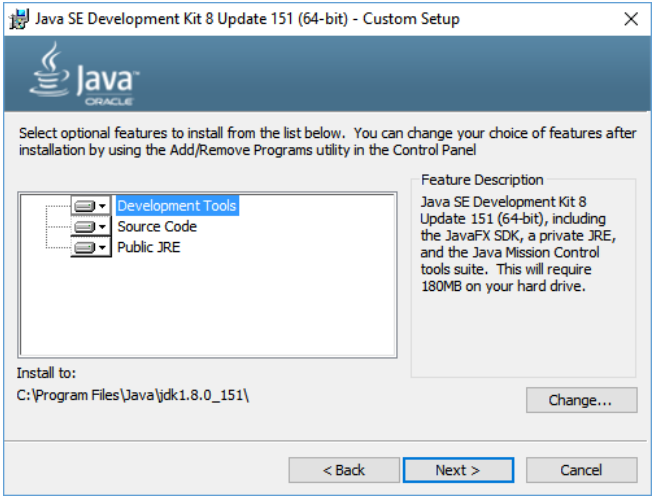
- Java **source code** files (**.java**) are compiled to Java **byte code** (**.class**) files
- **Byte code** is **interpreted** on the target platform within a **Java Virtual Machine**



5 Developing a Java Program
- Download and Install JDK
- Setting PATH and CLASSPATH
- Compiling and Running
- Your First Program: HelloWorld!

Download and Install JDK

- **Download JDK:** oracle.com/technetwork/java/javase/downloads/index.html
- **Install and take a note of JDK installation folder path**



Setting PATH and CLASSPATH

- **PATH** environment variable is used to specify the location of `javac`
- **CLASSPATH** environment variable is used to specify the location of user classes
- **Setting PATH:** In Windows, Control Panel > System > Edit Environment Variables > Edit or Add

  ```
  Variable Name: PATH
  Variable Value: C:\Program Files\Java\jdk1.8.0_151\bin
  ```

- **Setting CLASSPATH:** Follow the same procedure as the PATH environment variable, however classpath is the path where your .java files are saved. For example, in the Document folder (i.e. C:\Users\Teacher\Documents)

See figure 2 in next slide →

Setting PATH and CLASSPATH *(Cont'd)*

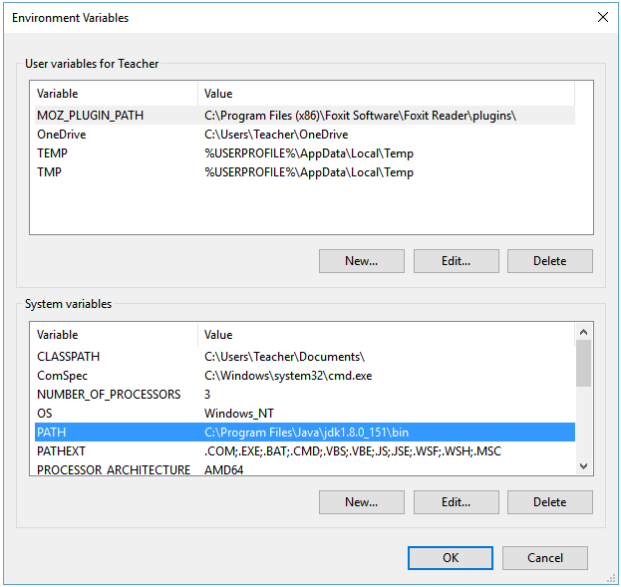


Figure 2: Setting up Java Environment Variables

Compiling and Running a Java Program

Compiling:

```
javac HelloWorld.java
```

Running:

```
java HelloWorld
```

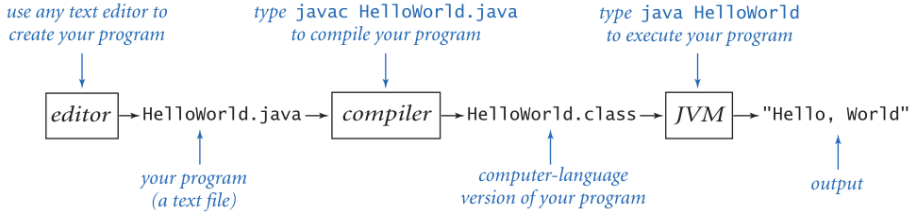


Figure 3: Writing, Compiling and Running a Java Program
*(**Figure source: Introduction to Programming in Java** by Robert Sedgewick & Kevin Wayne, 1st Edition, Pearson Education. p. 5)*

Your First Program: HelloWorld!

1. **Write** the following java program in a text editor i.e. Notepad++ (Windows) or Gedit (Linux) or TextEdit (Mac)

```
1 public class HelloWorld {
2
3     public static void main(String[] args) {
4         System.out.println("Hello World! \nHappy
              New Year 2018!");
5     }
6 }
```

2. **Compile** (in command-line):

```
> javac HelloWorld.java
```

3. **Run** (in command-line):

```
> java HelloWorld
```

Your First Program: HelloWorld!

1. **Write** the following java program in a text editor i.e. Notepad++ (Windows) or Gedit (Linux) or TextEdit (Mac)

```
1 public class HelloWorld {
2
3     public static void main(String[] args) {
4         System.out.println("Hello World! \nHappy
              New Year 2018!");
5     }
6 }
```

Program Output:

```
Hello World!
Happy New Year 2018!
```

2. **Compile** (in command-line):

```
> javac HelloWorld.java
```

3. **Run** (in command-line):

```
> java HelloWorld
```

Another Example

Note: If your Java program was saved in Documents folder for example, you would have the command line compilation and program execution similar to below -

```
C:\Users\Teacher\Documents>javac MySecondJavaProgram.java
C:\Users\Teacher\Documents>java MySecondJavaProgram
Hello World!
```

6 Editors / IDEs
Netbeans

Java Program Editors / IDEs

Popular Editors:

- Any text editor (i.e. Notepad++, JEdit, Gedit, TextEdit etc)

Popular IDEs:

- JCreator (jcreator.org/download.htm)
- Netbeans (netbeans.org/downloads)
- Eclipse (eclipse.org/downloads/eclipse-packages)
- IntelliJ IDEA (jetbrains.com/idea/download)

Netbeans

Netbeans:

- Free, Open-source, official Java IDE

Few Terms:

- **Run (F6)** - To compile and executed a Java project
- **Debug (Ctrl+F5)** - Special kind of running where programmers can set berak points, jump into certain code sections, check and modify and follow the flow of the code
- **Build (F11)** - To prepare a project / software for deployment (i.e. build source code to `jar` or `war`)

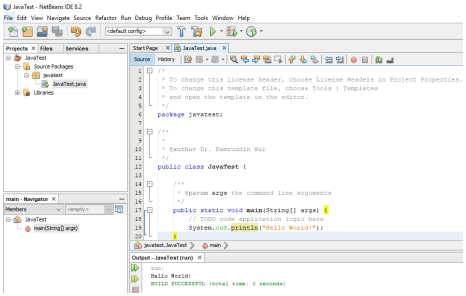


Figure 4: Netbeans IDE

7 Components of Java SE 8

Components of Java SE 8

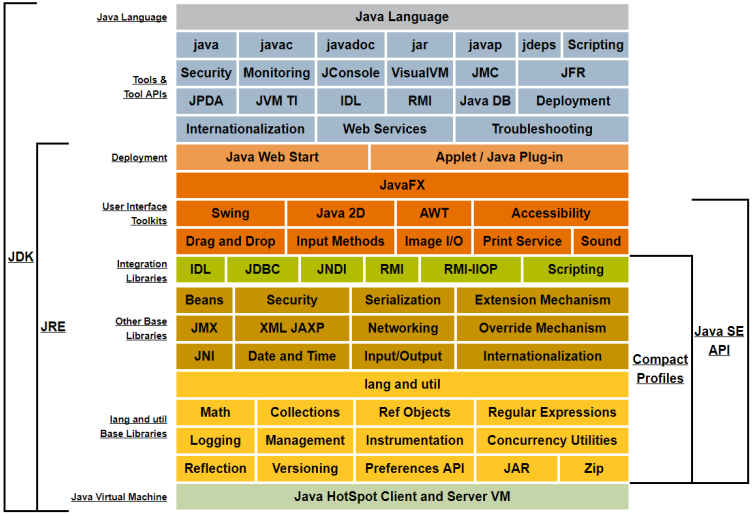


Figure 5: Java Platform Standard Edition 8 Documentation (Source: docs.oracle.com/javase/8/docs)

8 Java Packages

Java Packages

- In Java **Class library** or Application Programming Interface (**API**), classes are grouped into logical units called “Packages”
- Packages provide a mechanism for classifying (grouping) classes to learn and use
- Java Platform, Standard Edition 8 API Specification - docs.oracle.com/javase/8/docs/api/index.html
- Most commonly used packages are -

Language (general)	java.lang	Common classes used for all application development
GUI	java.awt java.awt.event javax.swing	Graphical User Interface, Windowing, Event processing
Misc. Utilities and Collections	java.util	Helper classes, collections
Input/Output	java.io	File and Stream I/O
Networking	java.net	Sockets, Datagrams

Thanks

Thanks for your time and attention!

kamruddin@aiub.edu
sites.google.com/view/kamruddinnur