Systems Software

COMP20081

Lecture 11 – Introduction to Java Programming

Dr Michalis Mavrovouniotis
School of Science and Technology
ERD 200
Office Hours: Thursday 12:00-14:00

NOTTINGHAM TRENT UNIVERSITY

Lecture Overview

- Overview of Java
- Simple comparison with C++
- Simple Java program

History and Philosophy of Java

- Developed by James Gosling in the early 90s.
  - History of Java: https://www.youtube.com/watch?v=8iytDbZ_fhA
- Designed with a C/C++ syntax style
- Strictly object-oriented language
- High-level programming language
- Highly portable – *"Write Once, Run Anywhere"*

High-Level Languages

- Machine code: the only language computers understand
- Assembly language: mnemonics for machine code instructions
- High level languages: provides us the ability to write programs that are easier to understand
- Compiler is a program that converts the programs written in a high level language to machine code

```
1110001100000000
0101011011100000
0111010001000000
0000100000001000
0001011011000100
0001001001100001
0111010001000000
```

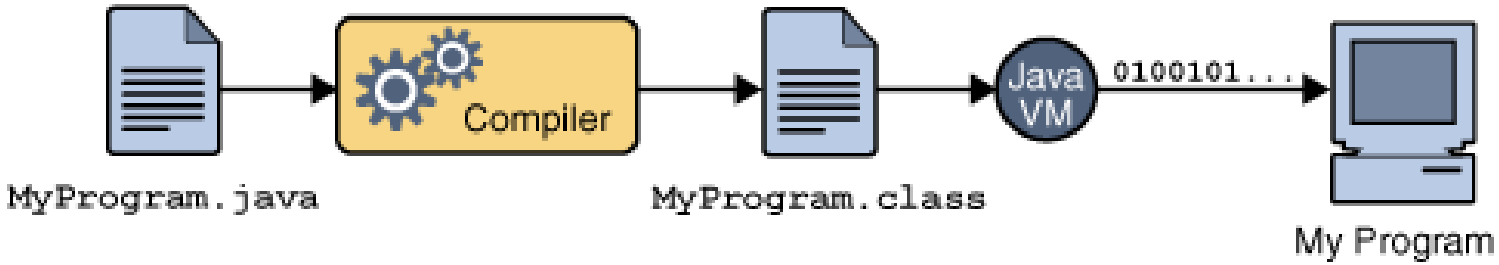
```
CMP AX,97
JL DONE
CMP AX,122
JG DONE
SUB AX,32
```

```
int x, y, sum;
x = 2;
y = 1;
sum = x + y;

return sum;
```

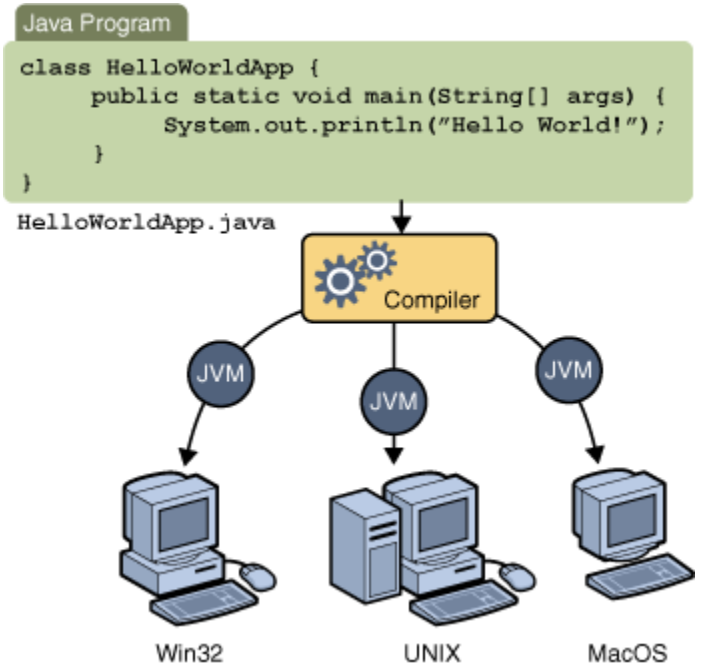
Java Language

- All source code is written in plain text with .java extension
- Compilation generates .class file that contains **bytecodes**
- The language of the Java Virtual Machine (JVM).



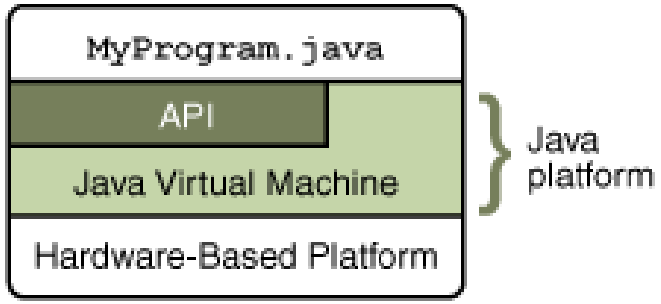
Java Virtual Machine

- JVM is available on many different operating systems/platforms
- The same .class is capable of running on multiple platforms
- JVM acts as an interpreter



The Java Platform

- The Java platform (environment) is a software-only platform that runs on top of other hardware-based platforms.
- Consists of two components:
  - JVM
  - Java application programming interface (API) libraries
- API is a large collection of ready-made components
  - http://docs.oracle.com/javase/8/docs/api/
- Grouped into libraries/classes
- Drawback ➔ Slower than C/C++



Java vs C++

- **Similarities**
  - Basic syntax: block structure { } (), semi-colon, statements
  - Import libraries (using import <> instead of #include <>)
- **Differences**
  - Java's compiler is more sophisticated, no need of a makefile
  - No header files
  - No pointers
  - Compiler generates bytecode and not machine code

Why Java?

- Well document ➔ javadoc documentation tool
  - http://docs.oracle.com/javase/8/docs/api/
- Easy to learn for programmers already familiar with C/C++
- Write less code
- Automatic garbage collection
- No platform dependencies
- Supports multi-threading and internetworking applications
- Library for GUI

Compile and Execute

- Manual way:
  - To compile a Java program: **javac MyProgram.java**
  - To execute the Java program: **java MyProgram**
  - If a program consists of multiple classes then compile only the modified
- Automatic way:
  - Use editors with an Integrated Development Environment (IDE).
    - NetBeans, Eclipse and many others.

First Java Program

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println(“Hello World!”); //Display the message
        }
        /*implement other methods here*/
}
```

Source Code Comments

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display the message
        }
        /*implement other methods here*/
}
```

Class Definition

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display the message
        }
        /*implement other methods here*/
}
```

Main Method

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display the message
        }
         /*implement other methods here*/
}
```

Main Method: Argument

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display the message
        }
        /*implement other methods here*/
}
```

Main Method: Statement

HelloWorldApp.java

```
/**
*Note that the filename should be
*the same with the class name
*/
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display the message
        }
         /*implement other methods here*/
}
```

Another Java Program

HelloWorldApp2.java

```
class HelloWorldApp2 {
        public static void main(String[] args) {
                for(int i = 0; i < 5; i++){
                        System.out.println(“Hello World!”);
                }
        }
}
```

And Another Java Program

HelloWorldApp3.java

```
/*Use of arguments*/
class HelloWorldApp3 {
        public static void main(String[] args) {
                if(args[0] == 1){
                        System.out.println(“Hello World!”);
                } else if (args[0] == 2) {
                        System.out.println(“Hello Universe!”);
                } else {
                        System.err.println(“error”);
                }
        }
}
```

And Another Java Program (correct)

HelloWorldApp3.java

```
/*Use of Java Documentation*/
class HelloWorldApp3 {
        public static void main(String[] args) {
                String temp = args[0];
                if(temp.equals(“1”)){
                        System.out.println(“Hello World!”);
                } else if (temp.equals(“2”)) {
                        System.out.println(“Hello Universe!”);
                } else {
                        System.err.println(“error”); }
        }
}
```

…And Another Java Program

HelloWorldApp4.java

```
/*Use of Java Documentation*/
class HelloWorldApp4 {
        public static void main(String[] args) {
                int temp = Integer.parseInt(args[0]);
                if(temp == 1){
                        System.out.println(“Hello World!”);
                } else if (temp == 2) {
                        System.out.println(“Hello Universe!”);
                } else {
                        System.err.println(“error”); }
        }
}
```

Summary

- Java Platform (API, JVM)
- Compilation and Execution (javac, java)