

Advanced Programming

Lecture 1: Introduction

Dr. Irfan Younas

January 12, 2016

Outline

1 Course Information

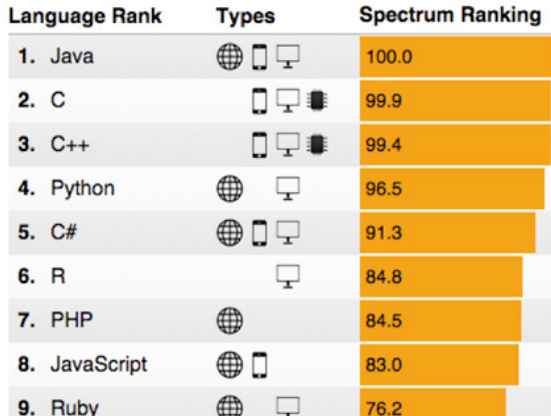
- Course Instructor and Office Hours

2 Introduction

- Top Programming Languages
- What is Java?
- History
- Characteristics of Java
- Building Standalone Java Programs
- Java Virtual Machine (JVM)
- Program Development in Java
- How Java differs from C?

- Course Instructor
 - Dr. Irfan Younas (PhD Computer Science)
Research Area: Genetic Algorithms, NP-hard Optimization, Combinatorial Optimization, Multi-objective Optimization, Machine Learning

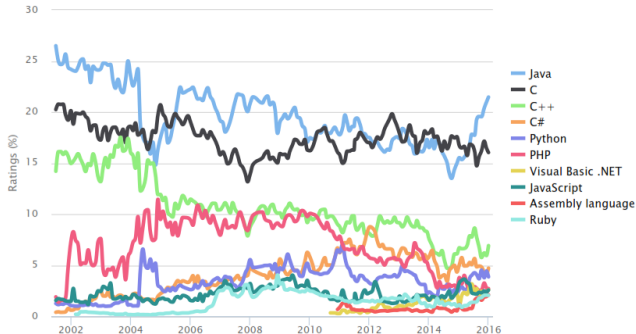
The 2015 Top Ten Programming Languages (IEEE Spectrum Ranking)



Top Programming Languages

TIOBE Programming Community Index

Source: www.tiobe.com



What is Java?

- Java is a **programming language** and a **platform**.
- Java is a high level, robust, secured and object-oriented programming language.
- **Platform:** Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

History

- Java is a programming language created by James Gosling from Sun Microsystems in 1991.
- The first publicly available version of Java (Java 1.0) was released in 1995.
- Java is an island of Indonesia where first coffee was produced (called java coffee).
- Sun Microsystems was acquired by the Oracle Corporation in 2010.

Characteristics of Java

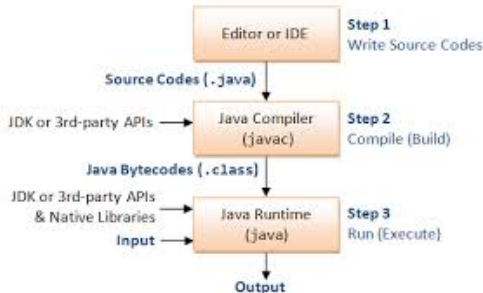
"Write Once and Run Anywhere."

- Simple
- Platform Independent
- Object-orientated
- Distributed
- Multi-Threaded
- Strongly-typed programming language
- Interpreted and compiled language
- Automatic memory management (Garbage Collection)

For details: visit <http://www.javatpoint.com/features-of-java>,
<http://www.vogella.com/tutorials/JavaIntroduction/article.html>

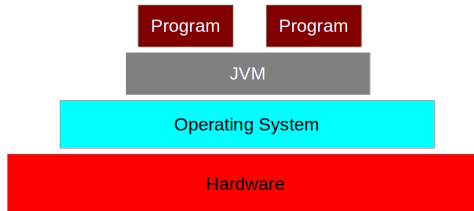
Building Standalone Java Programs

- Prepare the file `foo.java` using an editor
- Invoke the compiler: `javac foo.java`
- This creates `foo.class`
- Run the java interpreter: `java foo`



Java Virtual Machine (JVM)

- The .class files generated by the compiler are not executable binaries
 - so Java combines compilation and interpretation
- Instead, they contain byte-codes to be executed by the Java Virtual Machine
- This approach provides platform independence

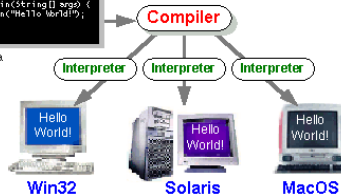




Java Program

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

HelloWorldApp.java



Program Development in Java

- Install the JDK (1.7 or 1.8)
- Write and edit the source code
 - Use Eclipse IDE for Java EE Developers. (for simple desktop programs you can use any editor, for instance windows notepad).
 - Source code file should have extension .java
- Compile the Source code (using javac)
- Run the program
 - java to run applications
 - appletviewer to run applets (embed applet in web page)

Program Structure in Java

- A java program typically consists of one or more classes
- Usually a separate class in separate file
- Each file is named after the class that is defined in it
- Extension of file .java

Example: HelloWorld.java

```
class HelloWorld {  
    //class definition  
}
```

Java Class

- In object-oriented programming, a class is a programming language construct used to group data and methods
- A method (called a function in some languages) is a set of instructions specific to a class.
- Object is a particular instance of a class

Java Class Library

- Collection of classes that facilitate your programs
- Set of classes are grouped into related sets called packages and each package is stored in separate directory
- A class in a package can access any of the other classes in the package
- A class in a package is named after its directory path e.g classes in java.lang are stored in java/lang
- The path of this directory is known to JRE
- `import java.io.*`

Some standard packages

| | |
|-----------|---|
| java.lang | Provides classes that are fundamental to the design of the Java programming language |
| java.io | Provides for system input and output through data streams, serialization and the file system |
| java.util | Contains the collections framework, legacy collection classes, event model, date and time facilities, internationalization, and miscellaneous utility classes |
| java.net | Provides the classes for implementing networking applications |
| java.awt | Contains all of the classes for creating user interfaces and for painting graphics and images |

Your First Java Program

- // My first program in Java

```
public class MyJava
{
    public static void main ( String args [ ] )
    {
        /* I am inside main . Execution always
        starts from method main in java applications
        */
        System.out.println ( "Hello JAVA" ) ;
    }
}
```

How Java differs from C?

- In C

```
int main (int argc, char** argv) {  
    printf("Hello World!\n");  
    return 0;  
}/* end main */
```

- In Java

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

- Exclusively Object-Oriented Language
 - EVERYTHING must live in a class (mostly)
 - No Global Variables
- No Pointers
 - No memory leaks (Auto Garbage Collection)

How Java differs from C?

- No Preprocessor (no **#include**, **#define**, etc.)
- No **goto** statement
- Declare/Define Variables & Methods anywhere (within a class)
- No **struct**, or **typedef**
- Cant overload Operators
- Use **new** rather than **malloc()**