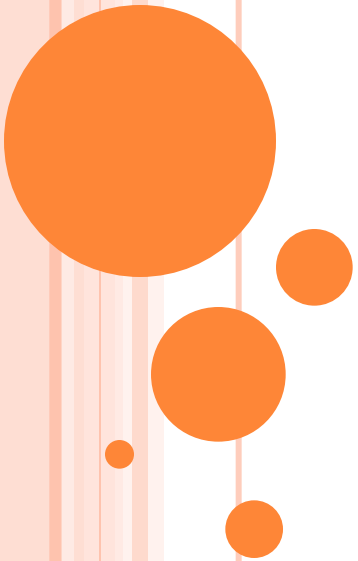


Introduction to OOP using Java

Tanjina Helaly

What is Programming?

- Instruction to computer/device to perform task.
- Computer understands only 0 and 1. Nothing else.
- So, we need to send the instruction in the form of 0, 1
  - Do you write program with just 0 and 1?

Classification/Evolution Of Programming

- Machine level programming
  - Send instruction in **binary** format
- Assembly Programming
  - send **code** instead of binary code.
  - Need **assembler** to convert to binary
  - "CSE 311 Microprocessor and Assembly Language"

Classification/Evolution Of Programming

- High level programming
  - Code is **close to English** Language
  - Need **Compiler** to convert to binary
  - 3 types
    - Non structured
    - Structured/Procedural
    - Object Oriented Programming

Classification/Evolution Of Programming

- Non structured
  - Generate spaghetti code
  - Sequential and has GoTo
  - COBOL, BASIC, FORTRAN
- Structured/Procedural
  - Use Subroutine/Function
  - improving the clarity, quality, and development time
  - C, PASCAL

Classification/Evolution Of Programming

- Object Oriented Programming
  - Object-oriented programming (OOP) is a programming language **model** organized around **objects** rather than "**actions**" and **data** rather than **logic**.
  - OOP focuses on **what (object)** developers want to manipulate rather than **how (logic)** they want to manipulate them.
  - Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.
  - Java, C++, C#

Programming Language

- A programming language is a formal constructed language designed to communicate instructions to a machine, particularly a computer.

Platform

- Programming languages can be platform dependent or independent.

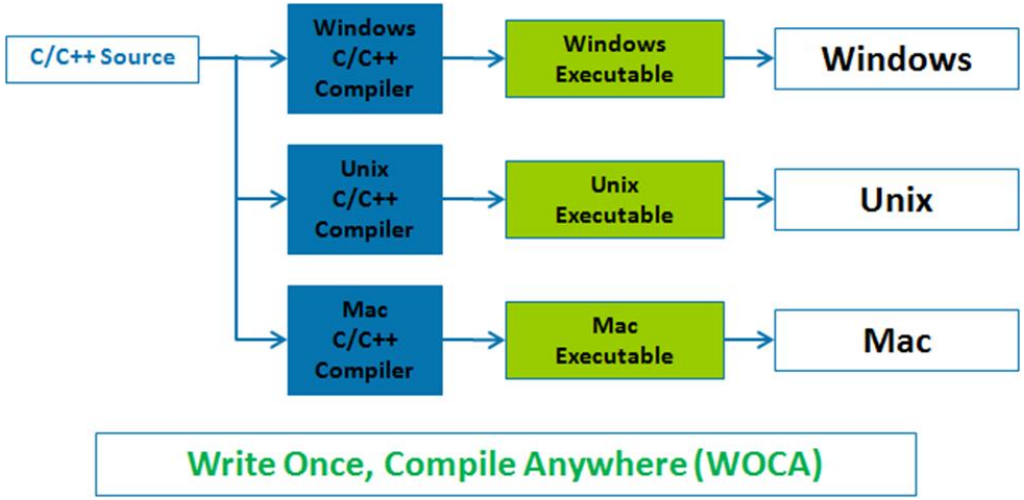
- What is platform?
  - Platform means the different types of computer that we use. For example :
    - Windows
    - Mac (Apple)
    - Linux

Platform

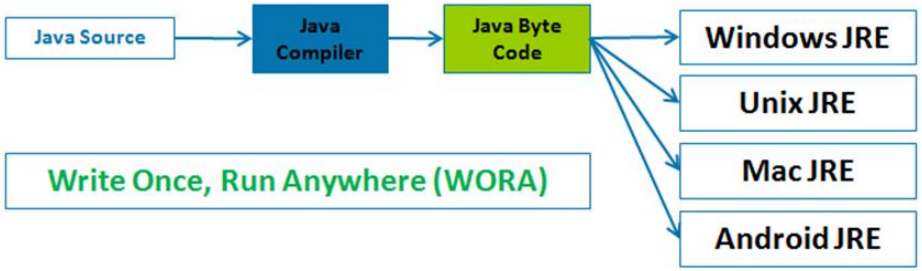
- C, C++, **Java**, Python, C# works on all these platforms
  - But for all of them require different compiler for different platform except java.
- Objective C programs works only on Mac

Regular Programming Languages vs. Java

- Regular Programming Languages



- Java



Our Goal

Learn Object Oriented Programming using Java

What is Java?

- A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language!
- Originally , it was targeted for digital cable television to make it interactive.
- Worked fabulous for Internet.
- Now it is everywhere.

JAVA - HISTORY

- Originally developed by **James Gosling** at **Sun Microsystems** (later acquired by **Oracle**)
  - James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991.
- Released in 1995
- Java was originally designed for interactive television,
  - but it was too advanced for the digital cable television industry at the time.
- Evolution of Name.
  - Oak->Green->Java

Java's Lineage

- Java is related to C++, which is a direct descendent of C.
  - Much of the character of Java is inherited from these two languages.
- From C, Java derives its syntax.
- Many of Java's object-oriented features were influenced by C++.

JAVA - CHARACTERISTICS

- Uses C/C++ basic syntax, basic data types (int, char, float, double, long, short, byte) and control structures
- "Pure" OO language
- No stand alone functions -**All code is part of a class**
- No explicit pointers - **uses references**
- Uses garbage collection

Java - Characteristics

- Java is strongly **typed**
- Java is normally compiled to a **bytecode**.
  - Java bytecode is a machine language for an abstract machine
  - Java bytecode is the instruction set of the Java virtual machine.
  - Makes Java secure and Portable
- Each platform (or browser) that runs Java has a Java Virtual Machine (JVM) . The JVM executes Java bytecodes

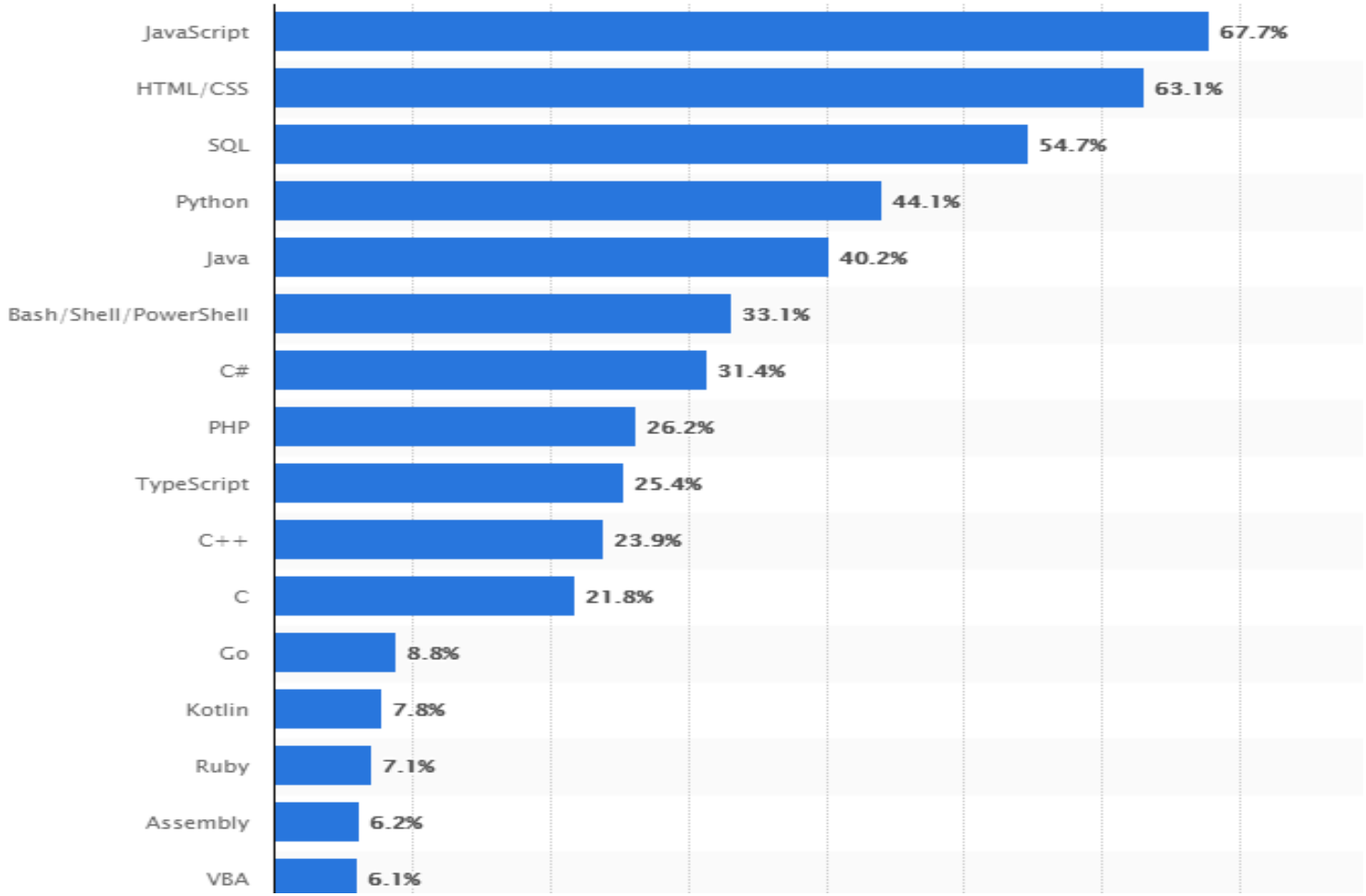
Java – The Platform

- Java has a large API (application programming interface) covering a wide range of areas .
- Java Foundation Classes (JFC) – GUI
- JDBC Database Access
- Java Web Server
- JavaBean
- Embedded Java - Java on embedded devices

Why Java?

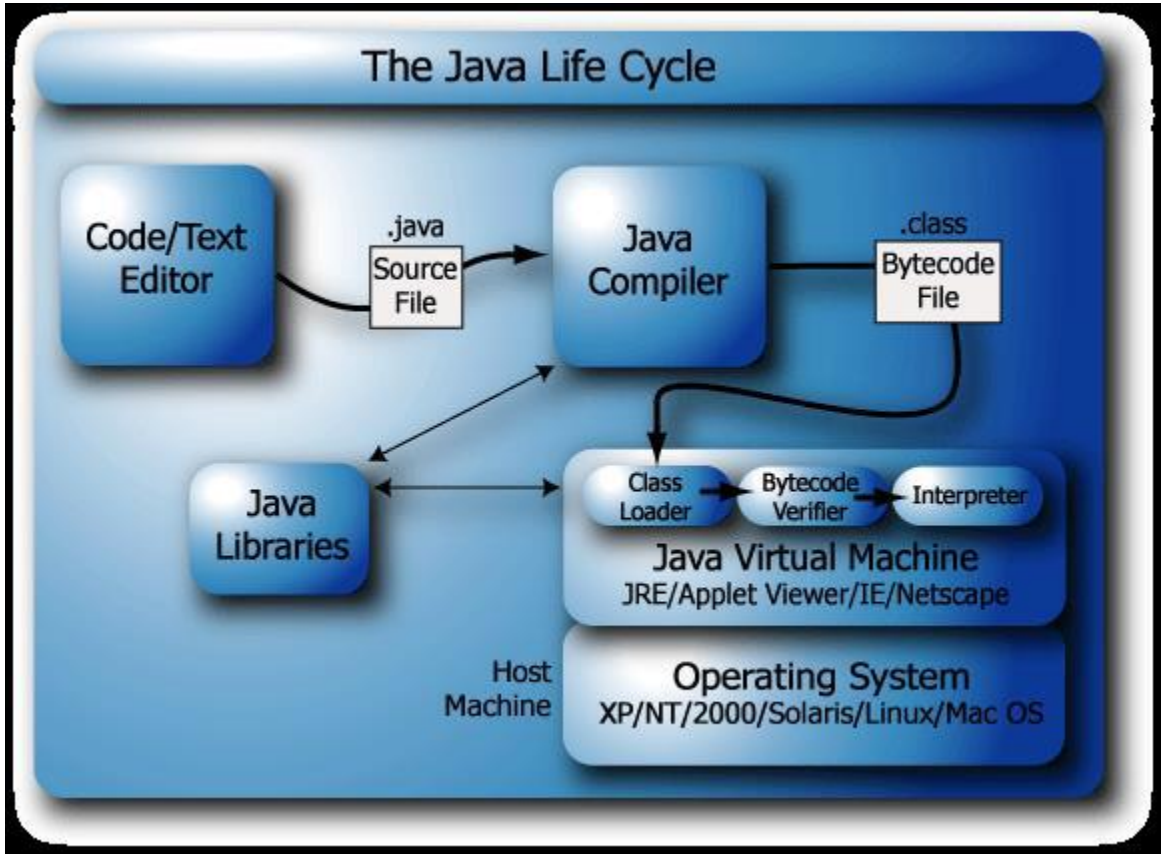
- Platform Independent - Code once run anywhere
  - Byte code
- Easy to learn
- Secure
  - Byte code & VM
- Free

Most used programming languages (as of early 2020)



- Source: https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

Java Life Cycle

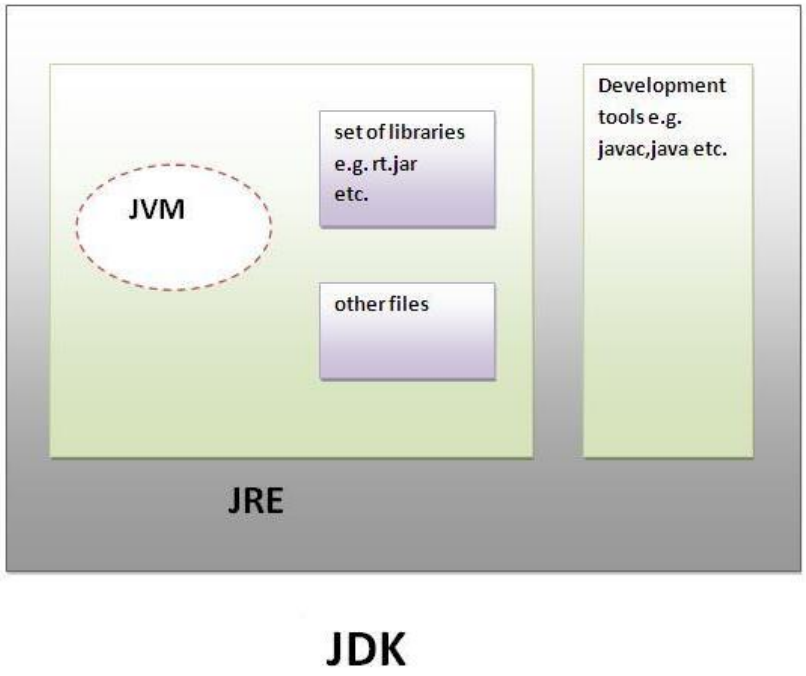


Tools

- JRE
  - JVM
- JDK

JRE AND JDK

- JRE:
  - JRE is an acronym for **Java Runtime Environment.**
  - It is used to provide runtime environment.
  - It is the implementation of JVM.
  - It physically exists. It contains set of libraries + other files that JVM uses at runtime.
- JDK
  - JDK is an acronym for **Java Development Kit**.
  - It physically exists.
  - It contains JRE + development tools



JVM

- **JVM (Java Virtual Machine)** is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.
- JVMs are available for many hardware and software platforms.
  - JVM, JRE and JDK are platform dependent because configuration of each OS differs.
  - But, Java is platform independent.
- The JVM performs following main tasks:
  - Loads code
  - Verifies code
  - Executes code
  - Provides runtime environment

Java Editions

- Java Standard Edition:
  - used for developing desktop applications and networking applications
- Java Enterprise Edition
  - used for developing large scale, distributed networking applications and web-based applications
- Java Micro Edition
  - used for developing applications for resource-constrained embedded devices

JAVA IDE

- Creating, Compiling, Debugging and Execution for these four steps JDK is not user friendly. IDE is provided for that. A list of IDEs are:
  - Eclipse
  - Netbeans.
  - IntelliJ IDEA
- In most of the IDE, the file will be compiled as soon as we update the file.
- Each IDE has menu, button and shortcut to run the application.

An Example HelloWorld

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

Java Source Code Naming Conventions

- All java source file should end with .java
- Each .java file can contain **only one public class**
- The **name of the file** should be **the name of the public class** plus ".java"
- Do not use abbreviations in the name of the class
- If the class name contains **multiple words** then **capitalize the first letter of each word** ex. HelloWorld.java

Naming Convention

- *Class Naming*
  - *Uses Capitalized word(s) i.e. Title case*
  - Examples:- HelloWorld, MyList, StudentMark
- *Variable and method names*
  - starts with a lowercase letter and after that use Title case
  - Examples:- variableAndMethodNames, aFloat, studentName
- *Names of constants*
  - All are capital letters and separated by underscore.
  - Example: NAMES_OF_CONSTANTS

Java Identifiers Rules

- Identifier is a name given to a variable, class, or method.
- Java identifier rules
  - Can contain letter, number, underscore (_), or dollar sign ($).
  - Cannot start with number. [Ex. 211csi (compile error)]
  - Identifiers are case sensitive. [test and Test are 2 different identifier]
  - have no maximum length.
  - cannot be a keyword, but it can contain a keyword as part of its name. [public- wrong but publicVar - OK]