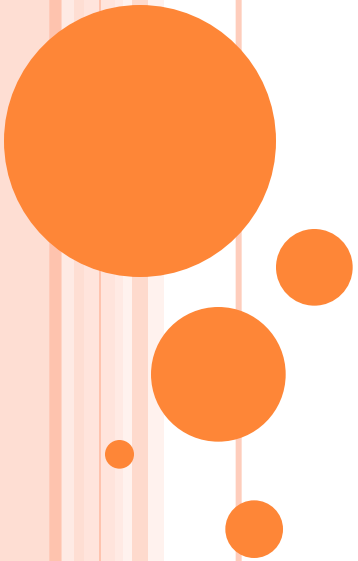


Introduction to OOP using Java

Tanjina Helaly

Programming & Programming Language

- Programming is instruction to computer/device to perform task.
- A programming language is a formal constructed language designed to communicate instructions to a machine

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
- 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
- Object Oriented Programming
  - Object-oriented programming (OOP) is a programming language model organized around objects rather than "actions" and data rather than logic.
  - Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.
  - Java, C++, C#

Our Goal

Learn Object Oriented Programming using 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 and basic data types -int, char, float, double, long, short, byte etc.
- Uses standard C/C++ 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 is strongly typed
- Java is normally compiled to a bytecode.
  - Java bytecode is a machine language for an abstract 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 The following list of Java APIs and applications from Sun show the range of applications of Java .
  - For reference http://java.sun.com/products/index.html
- Java Foundation Classes (JFC) – GUI
- JDBC Database Access
- Java Web Server
- EmbeddedJava - Java on embedded devices

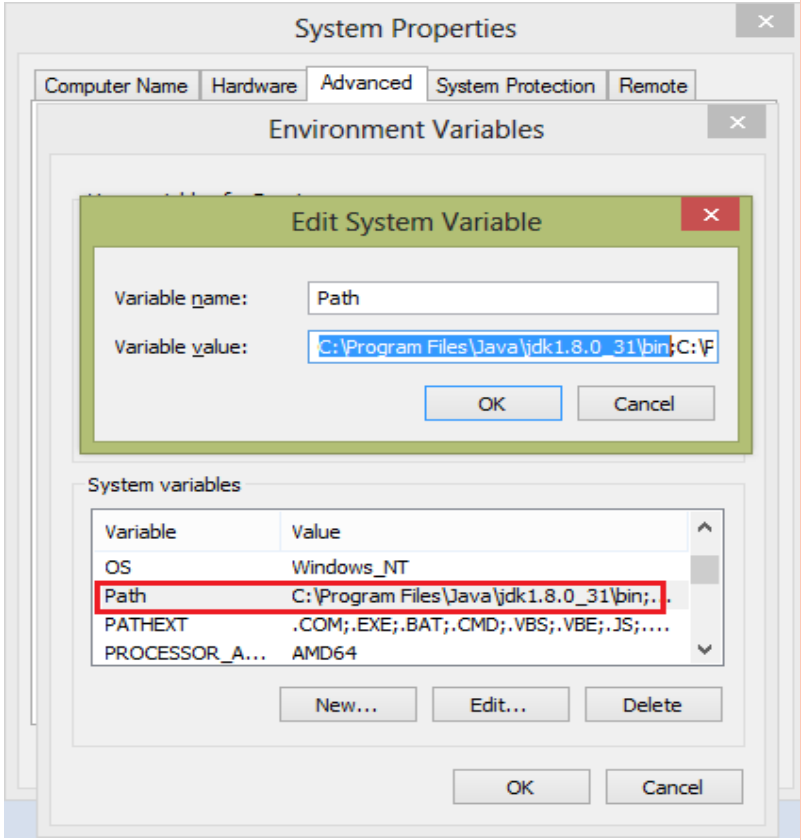
Why Java

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

Tools/Set-Up

STEP1: INSTALL JAVA AND PATH SET-UP

- Need to install Java(JDK and JRE). Get the latest version from Java Standard Edition(SE) from http://www.oracle.com/technetwork/java/javase/downloads/index.html
  - After installing Java you need to set-up the "**Path**" environment variable which is available from **My Computer** under **Advanced Properties** tab.
  - **Note**: Do not delete anything in "Path" variable. Just add your path "C:\Program Files\Java\jdk1.8.0_31\bin;" (Depending on your version the path will change) at the beginning of the existing value.



Step 2: Install IDE

- Need an IDE: Eclipse or NetBeans or IntelliJ IDEA.

  Or

- A Text Editor e.g. TextPad
- ***You can install***
  - eclipse from : http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/mars1
  - NetBeans: http://www.oracle.com/technetwork/java/javase/downloads/index.html
  - IntelliJ IDEA: https://www.jetbrains.com/idea/download/#section=windows

Compile & Run Java Application

Without IDE

- Using JDK you can compile and run java program from command line.
  - c:> javac HelloWorld. Java
    - compiling here and
    - it will produce HelloWorld.class i.e. bytecode.
  - c:>java HelloWorld
    - It runs java byte code on native machine

With 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

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

Data Types- Primitive Types

- In Java technology, data are divided into two broad categories: primitive types and class/reference types.
- Primitive data are eight types in four categories:
  - Logical: boolean (true or false)
  - Textual: char (16 bits)
  - Integral: byte (8 bits), short (16 bits), int (32 bits), and long (64 bits)
  - Floating point: oat (32 bits) and double (64 bits)
- Class or reference data used to create objects which are two types:
  - Textual: String
  - All classes that declare by yourself

Operator

- Assignment =
- Arithmetic + - * / %
- Equality == !=
- Relational < <= > >=
- Logical &&, | |
- increment/decrement ++ --
- Shift << >>

Control Statement

- if –else
- switch
- Loop
  - for
  - while
  - do-while

An Example HelloWorld

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

Reference

- Java:Complete Reference Chapter 1-5