Introduction to OOP using Java

Shekh. Md. Saifur Rahman
Lecturer, Department of CSE,
United International University
Email id : saifur@cse.uiu.ac.bd

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
- 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

Programming Language

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

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

Why Java

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

JAVA 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

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

You can install IntelliJ IDEA using the tutorial:

How to Install Intellij IDEA on Windows 10 + Creating First Hello World Java Application (youtube.com)

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
  - Can contain letter, number, underscore (_), or dollar sign ($).
  - Cannot start with number.
  - Identifiers are case sensitive(var and Var both can be declared in a program)
  - have no maximum length.
  - cannot be a keyword, but it can contain a keyword as part of its name.

- Write down whether the following identifiers are valid or not

Name	Valid/invalid	comment
myVar#		
myVar$		
$myVar		
final		
static		
finalVar		
1num		
main		

- Write down whether the following identifiers are valid or not

Name	Valid/invalid	comment
myVar#	invalid	# is not allowed
myVar$	valid	
$myVar	valid	
final	invalid	keyword
static	invalid	keyword
finalVar	valid	Can contain keyword as a part of name
1num	invalid	Cannot starts with number
main	valid	Not a keyword

Thank You