Programming in Java

Introduction

Java Features

- Java
  - no pointers
  - Is interpreted (C/C++ are Compiled)
  - No Preprocessor
  - No #define, #ifdef, #include, …
  - Concurrent
  - Lots of Librraries
  - Internet applications
  - Runs on the client side
  - Portable
  - Secure
  - Event Driven
  - Easy to Learn

Example

- Hello Class Program

```
public class Hello { // From JEIN
  public static void main(String argv[]) {
      System.out.println("Hello Class\n");
    System.exit(0);
  }
}
```

- **`main`** has a return type of **`void`** (not **`int`**)
  - The **`System.exit`** method is used to return value back to OS
- System.out.println is a print statement.

Compile/Execute

- The file name should be same as class name
- There can be multiple classes in the same file (For now, let us consider one class per file)
- Class is same as an Object
- javac <filename> compiles the java code File name has a .java extension eg. Hello.java
- It produces a class file (contains java byte code for Java Virtual Machines JVM). Eg. Hello.class
- java <filename without class> executes the program

Other Utilities

- Javap -package java.lang.Integer lists the methods and variables that are available in the package java.lang.Integer.
- Javap -c <classname. Produces a byte code of your program. Bytecode is written in Java Virtual Machine.
- Javadoc <filename> produces a HTML file which is a documentation of your program. One can see the documentation using a browser.

Names and Types

- variables
- functions, methods
- classes or Objects
- Types of variables - int, float, double, Boolean
- Arrays (unlike C or C++, in Java arrays are treated as an Object.)
- Life time of a variable

Types of Methods and Variables

- Instance variable. Instance methods
- Static variables and Static Methods
- public, private and protected variables and methods
- Constructor Method
- Automatic Variables

Import

- Import Statement
  - Without an import statement

    ```
    java.util.Calendar c1;
    ```
  - After the import statement

    ```
    import java.util.Calendar;
    ...
    Calendar c1;
    ```
  - Saves typing

    ```
    import java.util.*;  // Imports all classes
    ```

Examples

```
Public fact {
public static int factorial(int n) {
if (n==0)  return 1 else return n * factorial(n-1);
}
public static void main(String argv[]) {
int x;
x=9;
System.out.println("Factorial of"+x+"is"+factorial(x));
}
```


Expressions

- Arithmetic expressions in Java are similar to C/C++
  - Example

```
int i = 5 + 12 / 5 - 10 % 3
```

= 5 + (12 / 5) - (10 % 3)
= 5 + 2 - 1
= 6

  - Operators cannot be overloaded in Java
  - Integer division vs. floating point division
  - Operator precedence

Objects

- Objects
  - Instances of classes are called objects
  - Object variables store the *address* of an object
    - Different from primitive variables (which store the actual value)
    - Primitive Data Type example

```
int i=3;
int j=i;
i=2;  // i==2; j==3
```

    - Object Example1

```
java.awt.Button b1 = new java.awt.Button("OK");
java.awt.Button b2 = b1;
b2.setLabel("Cancel"); // Change is visible via b1 also
b1 = new java.awt.Button("Cancel")
```

  - No explicit dereferencing (i.e., no **&**, ***** or **->** operators)
    - No pointers
    - **null** = "Absence of reference" = a variable not pointing to an object