

3. Fundamentals of Java Programming

Lesson 1: Structure of a Java program

3.1. Structure of a Java program

3.1.1. Classes and Objects

- Java is an “Object Oriented” Programming Language. “Object Oriented” Programming Languages represent real world in templates known as classes. Classes do not represent the actual real world entities, but define characteristics and behavior of the real world entities.
- The actual real-world entities are represented by objects. Objects are instances of classes.

- Recall the “HelloWorld” program

```
class HelloWorld
{
    public static void main (String[] arguments)
    {
        System.out.println (“Hello World!!!”);
    }
}
```

- A Java Program is it self a real world entity. We define a template for it using the class:

```
class HelloWorld
```

3.1. Structure of a Java program

3.1.2. Attributes and Methods

- Real world entities have various characteristic attributes.
 - For example, a Person has a name, age, gender etc.
- Real world entities have characteristic behaviors.
 - For example, a Person may Walk, Talk, Sleep etc.
- Similarly, Objects in java also have attributes and perform methods. These are defined in the classes.
- “HelloWorld” has one method “main”. This is what is executed when we execute the program.

```
public static void main (String[] arguments)
{
    System.out.println (“Hello World!!!”);
}
```
- Since “HelloWorld” is a very simple program, it does not have any attributes. But we will come across classes with attributes later on.

3.2. Statements and expressions

3.2.1. Statements

- A method is made up of statements
- A statement is a simple command written in a programming language that causes something to happen.
 - For example, `System.out.println (“Hello World!!!”);`, prints “Hello World!!!” on the command line.
- In Java, we denote the end of a statement with a semicolon “;”.

3.2. Statements and expressions

3.2.2. Expressions

- As in mathematics, an “expression” in Java is a segment of code that can be evaluated to give a value.
- An expression is a combination of Variables, Literals, Operators and Functions (methods that return values).
- Expressions can be combined to give more complex expressions.

3.3. Comments in Java

- Comments are additional text included in programs, strictly for the benefit of the person reading the program. Comments are not compiled or executed.
 - For example, we might comment out “HelloWorld” program as:

```
/**
 @author John Smith
 @date 2007 January 1
 */

//This is my first program
class HelloWorld
{
    /*This method prints “Hello World!!!”
    to the command line*/
    public static void main (String[] arguments)
    {
        System.out.println (“Hello World!!!”);
    }
}
```

3.3. Comments in Java

3.3.1. Types of Comments

- There are three types of comments allowed in Java
 - Single-line comments
 - Single-line comments cover single-line or part of a single line. These begin with a “//”
 - For example,
`//This is my first program`
 - Multi-line comments
 - Multi-line comments may cover multiple lines. These begin with a “/*” and end with a “*/”.
 - For example,
`/*This method prints “Hello World!!!”
to the command line*/`
 - Javadoc comments
 - Javadoc comments are special multi-line comments, that can be used to generate java documentation using the javadoc tool. These begin with a “/**” and end with a “*/”.
 - For example,
`/**
@author John Smith
@date 2007 January 1
*/`

3.4. Literals

- A literal is a number, text or any other representation that directly represents its value.
 - For example, “Hello World!!!” is a literal that represents the string “Hello World!!!”.
- In Java, we have four types of literals
 - Number Literals (Integers and Real Numbers)
 - For example, 2004, 3.1416
 - Boolean Literals (Logical True and False)
 - For example, true, false
 - Character Literals (A single character)
 - For example, ‘1’, ‘a’
 - String Literals (A sequence or string of characters)
 - For example, “Hello World!!!”, “University of Colombo”