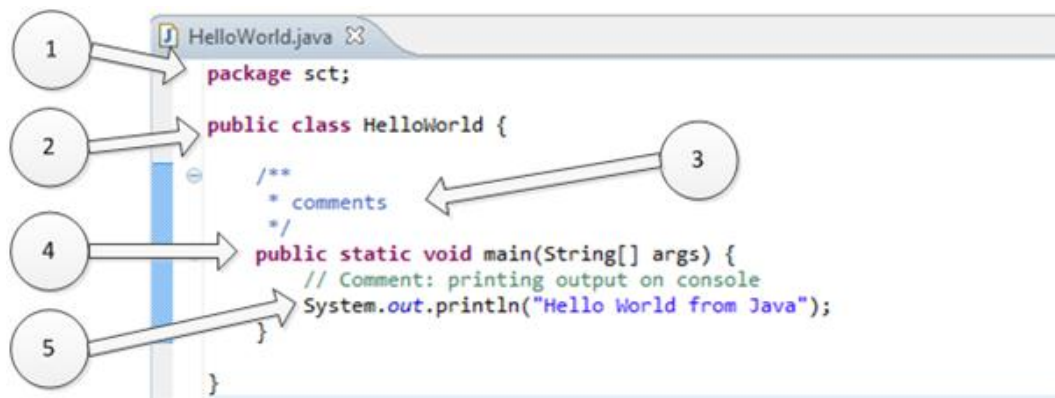


# JAVA PROGRAM STRUCTURE

## Description

---

Let's use example of HelloWorld Java program to understand structure and features of class. This program is written on few lines, and its only task is to print "Hello World from Java" on the screen. Refer the following picture.



### 1. "package sct":

---

It is package declaration statement. The package statement defines a name space in which classes are stored. Package is used to organize the classes based on functionality. If you omit the package statement, the class names are put into the default package, which has no name. Package statement cannot appear anywhere in program. It must be first line of your program or you can omit it.

---

## 2. “public class HelloWorld”:

---

This line has various aspects of java programming.

**a. public:** This is access modifier keyword which tells compiler access to class.

Various values of access modifiers can be public, protected,private or default (no value).

**b. class:** This keyword used to declare class. Name of class (HelloWorld) followed by this keyword.

## 3. Comments section:

---

We can write comments in java in two ways.

**a. Line comments:** It start with two forward slashes (//) and continue to the end of the current line. Line comments do not require an ending symbol.

**b. Block comments** start with a forward slash and an asterisk (/\*) and end with an asterisk and a forward slash (\*/\*).Block comments can also extend across as many lines as needed.

## 4. “public static void main (String [ ] args)”:

---

Its method (Function) named main with string array as argument.

**a. public :** Access Modifier

**b. static:** static is reserved keyword which means that a method is accessible and usable even though no objects of the class exist.

**c. void:** This keyword declares nothing would be returned from method. Method can return any primitive or object.

**d. Method content inside curly braces.** { } asdfla;sd

**5. System.out.println("Hello World from Java") :**

---

**a. System:**It is name of Java utility class.

**b. out:**It is an object which belongs to System class.

**c. println:**It is utility method name which is used to send any String to console.

**d. "Hello World from Java":**It is String literal set as argument to println method.

**More Information regarding Java Class:**

---

- Java is an object-oriented language, which means that it has constructs to represent objects from the real world. Each Java program has at least one class that knows how to do certain things or how to represent some type of object. For example, the simplest class, HelloWorld, knows how to greet the world.
- Classes in Java may have methods (or functions) and fields (or attributes or properties).

- Let's take example of Car object which has various properties like color, max speed etc. along with it has functions like run and stop. In Java world we will represent it like below:

view plainprint?

```
1. package sct;
2. public class Car {
3.     private String color;
4.     private int maxSpeed;
5.     public String carInfo(){
6.         return color + " Max Speed:-" + maxSpeed;
7.     }
8.     //This is constructor of Car Class
9.     Car(String carColor, int speedLimit){
10.        this.color = carColor;
11.        this.maxSpeed =speedLimit;
12.    }
13. }
```

- Lets make a class named CarTest which will instantiate the car class object and call carInfo method of it and see output.

view plainprint?

```
1. package sct;
2. //This is car test class to instantiate and call Car objects.
3. public class CarTest {
4.   public static void main(String[] args) {
5.     Car maruti = new Car("Red", 160);
6.     Car ferrari = new Car ("Yellow", 400);
7.     System.out.println(maruti.carInfo());
8.     System.out.println(ferrari.carInfo());
9.   }
10. }
```

Output of above CarTest java class is as below. We can run CarTest java program because it has main method. Main method is starting point for any java program execution. Running a program means telling the Java Virtual Machine (JVM) to "Load theclass, then start executing its main () method. Keep running 'til all thecode in main is finished."

```
Red-Make Year :1999 Max Speed :-160
Yellow-Make Year :2012 Max Speed :-400
```

## Common Programming guidelines:

---

- Java identifiers must start with with a letter, a currency character (\$), or a connecting character such as the underscore ( \_ ). Identifiers cannot start with a number. After first character identifiers can contain any combination of letters,currency characters, connecting characters, or numbers. For example,
  - `int variable1 = 10 ; //This is valid`
  - `int 4var =10 ; // this is invalid, identifier can't start with digit.`
- Identifiers, method names, class names are case-sensitive; var and Var are two different identifiers.
- You can't use Java keywords as identifiers, Below table shows list of Java keywords.

abstract	boolean	break	byte
char	class	const	continue
double	else	extends	final
for	goto	if	implements
int	interface	long	native
private	protected	public	return
strictfp	super	switch	synchronized

throws	transient	try	void
assert	enum		

- **Classes and interfaces:** The first letter should be capitalized, and if several words are linked together to form the name, the first letter of the inner words should be uppercase (a format that's sometimes called "camelCase").
- **Methods:** The first letter should be lowercase, and then normal camelCase rules should be used. For example:
  - getBalance
  - doCalculation
  - setCustomerName
- **Variables:** Like methods, the camelCase format should be used, starting with a lowercase letter. Sun recommends short, meaningful names, which sounds good to us. Some examples:
  - buttonWidth
  - accountBalance
  - empName
- **Constants:** Java constants are created by marking variables static and final. They should be named using uppercase letters with underscore characters as separators:

- MIN\_HEIGHT
- There can be only one public class per source code file.
- Comments can appear at the beginning or end of any line in the source code file; they are independent of any of the positioning rules discussed here.
- If there is a public class in a file, the name of the file must match the name of the public class. For example, a class declared as “public class Dog { }” must be in a source code file named Dog.java.
- To understand more about access modifiers applicable to classes, methods, variables in access modifier tutorial.

We will understand more about constructors, access modifiers in coming tutorials.

Source code of sample discussed attached here to run directly on your system.

## Summary

---

- Java program has first statement as package statement (if package declared).
- One Java file can have multiple classes declared in it but only one public class which should be same as file name.
- Java class can have instance variable such as methods, instance variable.

Source: <http://www.w3resource.com/java-tutorial/java-program-structure.php>