



BASIC PROGRAM STRUCTURE IN JAVA

MY FIRST PROGRAM IN JAVA

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



MY FIRST PROGRAM IN JAVA

○ Understanding first java program

- **class** keyword is used to declare a class in java.
- **public** keyword is an access modifier which represents visibility, it means it is visible to all.
- **static** is a keyword, if we declare any method as static, it is known as static method. The core advantage of static method is that there is no need to create object to invoke the static method.
- **void** is the return type of the method, it means it doesn't return any value.
- **main** represents startup of the program.
- **String[] args** is used for command line argument.
- **System.out.println()** is used print statement.



BASIC STRUCTURE OF JAVA PROGRAM



Structure of Java Program



DOCUMENTATION SECTION

- The documentation section is an important section but optional for a Java program.
- It includes **basic information** about a Java program.
- The information includes the **author's name, date of creation, version, program name, company name, and description** of the program.
- It improves the readability of the program.
- Whatever we write in the documentation section, the Java compiler ignores the statements during the execution of the program.
- To write the statements in the documentation section, we use **comments**. The comments may be **single-line, multi-line, and documentation** comments.

DOCUMENTATION SECTION

- **Single-line Comment:** It starts with a pair of forwarding slash (//).

For example:

```
//First Java Program
```

- **Multi-line Comment:** It starts with a /* and ends with */. We write between these two symbols.

For example:

```
/*It is an example of multiline comment*/
```

- **Documentation Comment:** It starts with the delimiter (/**) and ends with */.

For example:

```
/**It is an example of documentation comment*/
```



PACKAGE STATEMENT

- The first statement allowed in a java file is a package statement.
- This statement declare a package name and informs the compiler that the classes defined here belong to this package.

- Example :

```
package student;
```

The package statement is optional. That is, our classes do not have to be part of package.



IMPORT STATEMENT

- The next thing after a package statement (but before any class definition) may be a number of import statements.
- This is similar to the `#include` statements in C.
- *Import student.test;*
- This statement instructs the interpreter to load test class contained in the package.
- Using import statements, we can have access to classes that are part of other named packages.



INTERFACE STATEMENTS

- An interface is like a class but includes a group of method declarations.
- This is also an optional section and is used only when we wish to implement the multiple inheritance feature in the program.



CLASS DEFINITIONS

- A java program may contain multiple class definitions.
- Classes are the primary and essential elements of a java program.
- These classes are used to map the objects of real world problems.



MAIN METHOD CLASS

- Since every java stand alone program requires a main method as its starting point, this class is the essential part of a java program.



NAMING CONVENTION

○ Case Sensitivity

- Java is case sensitive
- Example: **Hello** and **hello** would have different meaning in Java.

○ Class Names

- For all class names the first letter should be in Upper Case.
- If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.
- Example

class MyFirstJavaClass



NAMING CONVENTION

Method Names

- All method names should start with a Lower Case letter.
- If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.
- Example

public void myMethodName()



NAMING CONVENTION

Program File Name

- Name of the program file should exactly match the class name.
- When saving the file, you should save it using the class name and append '.java' to the end of the name.
- If the file name and the class name do not match your program will not compile
- Example : Assume 'MyFirstJavaProgram' is the class name.
Then the file should be saved as 'MyFirstJavaProgram.java'

