

First JAVA program

Since Java is an Object-oriented language, it requires to write a code inside a class. Let us look at a simple java program.

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

Let's understand what above program consists of and its key points.

class : class keyword is used to declare classes in Java

public : It is an access specifier. Public means this function is visible to all.

static : static is again a keyword used to make a function static. To execute a static function, you do not have to create an Object of the class. The main() method here is called by JVM, without creating any object for class.

void : It is the return type, meaning this function will not return anything.

main : main() method is the most important method in a Java program. This is the method which is executed, hence all the logic must be inside the main() method. If a java class is not having a main() method, it causes compilation error.

String[] args : is used for command line argument. This represents an array whose type is String and name is args.

System.out.println : This is used to print anything on the console like printf in C language. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class.

Steps to Compile and Run your first Java program

Step 1: Open a text editor and write the code as above.

Step 2: Save the file as Hello.java

Step 3: Open command prompt and go to the directory where you saved your first java program assuming it is saved in C drive.

Step 4: Type `javac Hello.java` and press Return(Enter KEY) to compile your code. This command will call the Java Compiler asking it to compile the specified file. If there are no errors in the code the command prompt will take you to the next line.

Step 5: Now type `java Hello` on command prompt to run your program.

Step 6: You will be able to see Hello world program printed on your command prompt.

Valid Java main() method signature

1. `public static void main(String[] args)`
2. `public static void main(String []args)`
3. `public static void main(String args[])`
4. `public static void main(String... args)`
5. `static public void main(String[] args)`
6. `public static final void main(String[] args)`
7. `final public static void main(String[] args)`
8. `final strictfp public static void main(String[] args)`

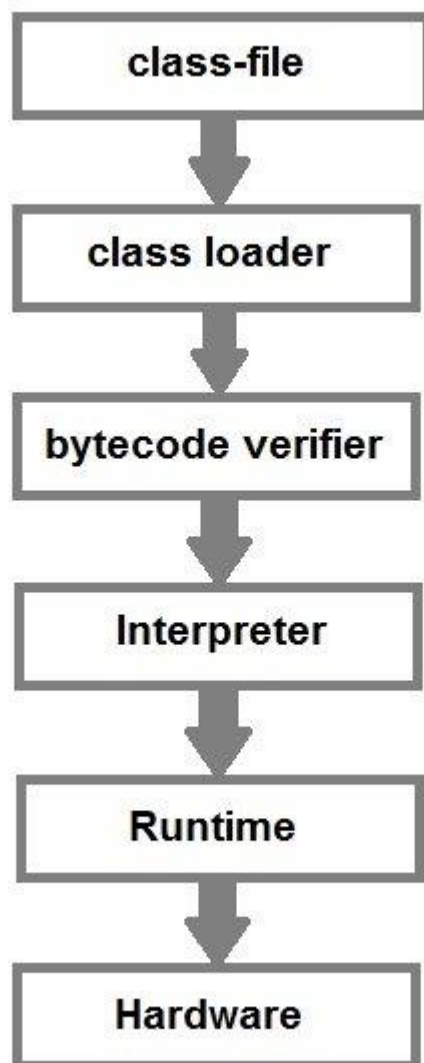
Invalid Java main() method signature

1. `public void main(String[] args)`
2. `static void main(String[] args)`
3. `public void static main(String[] args)`
4. `abstract public static void main(String[] args)`

Let us see what happens at Runtime

After writing your Java program, when you will try to compile it. Compiler will perform some compilation operation on your program.

Once it is compiled successfully byte code (.class file) is generated by the compiler.



After compiling when you will try to run the byte code (.class file), the following steps are performed at runtime.

Class loader loads the java class. It is subsystem of JVM Java Virtual machine.

Byte Code verifier checks the code fragments for illegal codes that can violate access right to the object.

Interpreter reads the byte code stream and then executes the instructions, step by step.