**public static void main(String[] args)**

Java main method is the entry point of any java program. Its syntax is always public static void main(String[] args). You can only change the name of String array argument, for example you can change args to myStringArgs.

Also String array argument can be written as String... args or String args[].

### public

This is the access modifier of the main method. It has to be public so that java runtime can execute this method. Remember that if you make any method non-public then it’s not allowed to be executed by any program, there are some access restrictions applied. So it means that the main method has to be public. Let’s see what happens if we define the main method as non-public.

### static

When java runtime starts, there is no object of the class present. That’s why the main method has to be static so that JVM can load the class into memory and call the main method. If the main method won’t be static, JVM would not be able to call it because there is no object of the class is present. Let’s see what happens when we remove static from java main method.

### void

Java programming mandates that every method provide the return type. Java main method doesn’t return anything, that’s why it’s return type is void. This has been done to keep things simple because once the main method is finished executing, java program terminates. So there is no point in returning anything, there is nothing that can be done for the returned object by JVM. If we try to return something from the main method, it will give compilation error as an unexpected return value. For example, if we have the main method like below.

### main

This is the name of java main method. It’s fixed and when we start a java program, it looks for the main method. For example, if we have a class like below.

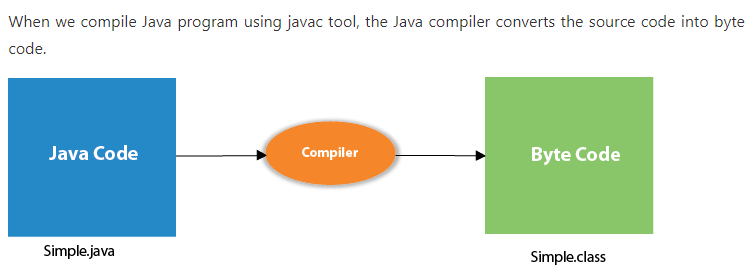
### String[] args

Java main method accepts a single argument of type String array. This is also called as java command line arguments. Let’s have a look at the example of using java command line arguments.

Sample Programs:

### **Creating Hello World Example**

1. **class** Simple{
2. **public** **static** **void** main(String args[]){
3. System.out.println("Hello Java");
4. }
5. }



Let's see what is the meaning of class, public, static, void, main, String[], System.out.println().

* **class** keyword is used to declare a class in Java.
* **public** keyword is an access modifier that represents visibility. It means it is visible to all.
* **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method. So, it saves memory.
* **void** is the return type of the method. It means it doesn't return any value.
* **main** represents the starting point of the program.
* **String[] args** or **String args[]** is used for [command line argument](https://www.javatpoint.com/command-line-argument). We will discuss it in coming section.
* **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class. We will discuss the internal working of [System.out.println()](https://www.javatpoint.com/system-out-println-in-java) statement in the coming section.

### How to Print an Integer entered by an user

import java.util.Scanner;

public class HelloWorld {

public static void main(String[] args) {

// Creates a reader instance which takes

// input from standard input - keyboard

Scanner reader = new Scanner(System.in);

System.out.print("Enter a number: ");

// nextInt() reads the next integer from the keyboard

int number = reader.nextInt();

// println() prints the following line to the output screen

System.out.println("You entered: " + number);

}

}

### Printing Variables and Literals

class Variables {

public static void main(String[] args) {

Double number = -10.6;

System.out.println(5);

System.out.println(number);

}

}

### Print Concatenated Strings

class PrintVariables {

public static void main(String[] args) {

Double number = -10.6;

System.out.println("I am " + "awesome.");

System.out.println("Number = " + number);

}

}

### Get Integer Input From the User

import java.util.Scanner;

class Input {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.print("Enter an integer: ");

int number = input.nextInt();

System.out.println("You entered " + number);

// closing the scanner object

input.close();

}

}

### Get float, double and String Input

import java.util.Scanner;

class Input {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

// Getting float input

System.out.print("Enter float: ");

float myFloat = input.nextFloat();

System.out.println("Float entered = " + myFloat);

// Getting double input

System.out.print("Enter double: ");

double myDouble = input.nextDouble();

System.out.println("Double entered = " + myDouble);

// Getting String input

System.out.print("Enter text: ");

String myString = input.next();

System.out.println("Text entered = " + myString);

}

}