**Can we overload main() method in Java?**

Yes, We can overload main() method. A Java class can have any number of main() methods. But to run the java class, class should have main() method with signature as **“public static void main(String[] args)”**. If you do any modification to this signature, compilation will be successful. But, you can’t run the java program. You will get run time error as **main method not found**.

1. public class MainMethod
2. {
3. public static void **main**(**String**[] args)
4. {
5. System.out.println("Execution starts from this method");
6. }
7. void **main**(**int** args)
8. {
9. System.out.println("Another main method");
10. }
11. **double** **main**(**int** i, **double** d)
12. {
13. System.out.println("Another main method");
14. **return** d;
15. }
16. }

**Can we declare main() method as private or protected or with no access modifier?**

No, main() method must be public. You can’t define main() method as private or protected or with no access modifier. This is because to make the main() method accessible to JVM. If you define main() method other than public, compilation will be successful but you will get run time error as no main method found.

1. public class MainMethod
2. {
3. private static void **main**(**String**[] args)
4. {
5. //Run time error
6. }
7. }

## Can We Declare main() Method As Non-Static?

No, main() method must be declared as static so that JVM can call main() method without instantiating it’s class. If you remove ‘static’ from main() method signature, compilation will be successful but program fails at run time

1. public class MainMethod
2. {
3. public void **main**(**String**[] args)
4. {
5. System.out.println(1); //Run time error
6. }
7. }

## Why main() method must be static?

## Suppose, If main() is allowed to be non-static, then while calling the main method JVM has to instantiate it’s class. While instantiating it has to call constructor of that class. There will be an ambiguity if constructor of that class takes an argument. For example, In the below program what argument JVM has to pass while instantiating class “MainMethod”?.

1. public class MainMethod
2. {
3. public **MainMethod**(**int** i)
4. {
5. //Constructor taking one argument
6. }
7. public void **main**(**String**[] args)
8. {
9. //main method as non-static
10. }
11. }

## Can we change return type of main() method?

## No, the return type of main() method must be void only. Any other type is not acceptable.

1. public class MainMethod
2. {
3. public static **int** **main**(**String**[] args)
4. {
5. **return** 1; //run time error : No main method found
6. }
7. }

## Can main() method take an argument other than string array?

## No, argument of main() method must be string array. But, from the introduction of var args you can pass var args of string type as an argument to main() method. Again, var args are nothing but the arrays.

1. public class MainMethod
2. {
3. public static void **main**(**String**... args)
4. {
5. //Var args as an argument
6. }
7. }

## Can we run java class without main() method?

## No, you can’t run java class without main method. But, there are some scenarios like if super class has main() method, then sub class can be run without defining main() method in it. For example, if you run class B in the below program, you will get **1** as output.

1. class A
2. {
3. public static void **main**(**String**[] args)
4. {
5. System.out.println(1);
6. }
7. }
8. public class B extends A
9. {
10. }