**Main method :** Java program execution starts with main method. If we run a program without writing main method then we get runtime exception : No such method error : main. So, the presence of main method is checked at run time by JVM not by compiler at compile time. JVM searches a main method with prototype : public static void main (String args[]). Inside, this syntax is configured, hence it always searches for this syntax only. **We can change main method name**, but for that customization of JVM is required. We won’t do it normally, we use main only.

Public : to call by jvm from anywhere, main method should be public.

Static : without any existing object, jvm has to call main method, so it is static. And main method is no were related to any particular object.

Void : main method won’t return anything to JVM hance return type is void.

Main : this is the name which is configured inside JVM

String args[] : command line arguments (arguments we are passing from command line)

The syntax of main is very strict and if we change it then we get runtime exception. Also, a few changes to the syntax are acceptable :

1. Order of modifiers is not important (we can have static public.. )
2. Valid array declaration is acceptable (String[] args / String []args / String args[])
3. We can have main method with final keyword

**1. Overloading** of main method is possible and JVM always calls 1st one of the following(because this is the expected syntax) :

Class Test

{

public static void main (String args[]) // overloading of main method

{

}

public static void main (**int** args[]) // overloading of main method

{

}

}

**2**. **Inheritance** concept is applicable for main method. While executing child class, if child class doesn’t contain main method then parent class main method will be executed.

Class Parent

{

public static void main (String args[])

{

}

}

Class child extends Parent

{

}

**3**.**Method overloading - hiding** : parent and child both contains public static void main(String args[]). Since main is static, this is known as method hiding that means parent and child posses exactly same static methods. The method in the super class will be hidden by the one that is in the sub class.