Eg: Main\_1 , Main\_In\_Class\_2 , Main\_Outside\_Class4

------------------------------------------------------------------------------------------------

* Program can have as many .class files a program consists of

---------------------------------------------------------------------------------------------------------

\*main method can be overloaded but jvm takes the only the one with regular format i.e (public static void main(String []args)

Eg: Main\_Method\_Overloading\_3

Default prototype

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

### Order of Modifiers

The static and public modifiers in the main() method declaration can be interchanged

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

### Variants of String array arguments

In Java, it is possible to place the square brackets at different positions

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

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

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

### Args or anything

 you can use any valid Java identifier instead of args

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

Command line arguments :

Eg: Command\_Line\_Arguments \_1

Eg: Command\_Line\_Arguments\_2

Command line Arguments starts from 0 index number . it is of type string

Eg: Command\_Line\_Arguments\_3

// program flow

1. First compiler will do type checking whether the arguments given is of String type
2. Here in 3rd line we have given “day” to add in the command line ,but during the runtime I have given the argument

java Good

1. Now the jvm shouts “ hey I have already allocated space for “Good” now I cant allocate space for day in program
2. So jvm throws array out of bonds exception

Give the command line arguments only at run time

Eg: Command\_Line\_Arguments\_4

// here I have given 2 arguments but printed only 0th index

Eg: Command\_Line\_Arguments\_5

args is the array and since array you can use length property on it . and args is an array of string ,so data present in array is always a string

we can apply equals method on string objects so there is no compile time error .

here we have given command line arguments only at 0th index but tried to access 1st index so array index out of bounds.

Eg: Command\_Line\_Arguments\_6

// go through the code

Eg: Command\_Line\_Arguments\_7

x is intialized in the block , there is no gaurantee that block would be executed by compiler , so it will throw compile time error . if there is no else block

Here the compiler understands since x is also initialized in else ,incase if block does not executes ,else block will execute compulsory and x will be initialized.

### Var-args instead of String array

you can use a varargs parameter (denoted by three dots ... instead of square brackets []) as a replacement.

**public** **static** **void** main(String... args)

### The final Modifier String argument

ensuring that the parameter's value cannot be changed.

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

### Final main method

possible to declare the main() method with the final keyword,

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

### Synchronized Keyword to Static Main Method

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

### strictfp keyword to the static main method

 strictfp keyword ensures consistent floating-point arithmetic across different platforms.

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

### Inheritance of the Main method

Eg: Inheritance\_Of\_Main\_Method\_6

### Method Hiding of main(), but not Overriding

A **static method cannot be overridden** as an instance method in Java and vice versa . since main is static it cannot be overridden. method overriding is done only by non static methods.

Eg: Hiding\_Of\_Main\_Method\_5

There should be only one public class .

-----------------------------------------------------------------------------------------

Class name and file name can be different .

Give file name for compilation .

Class name during run time

Don’t use public for class name in this program it wil show error.

JVM executes the static block, then it executes static methods, and then it creates the object needed by the program. Finally, it executes the instance methods. JVM executes a static block on the highest priority basis. It means JVM first goes to static block even before it looks for the main() method in the program.

Eg: Excuetion\_Process

Eg: SemiColon\_Example

// check Documentation in program