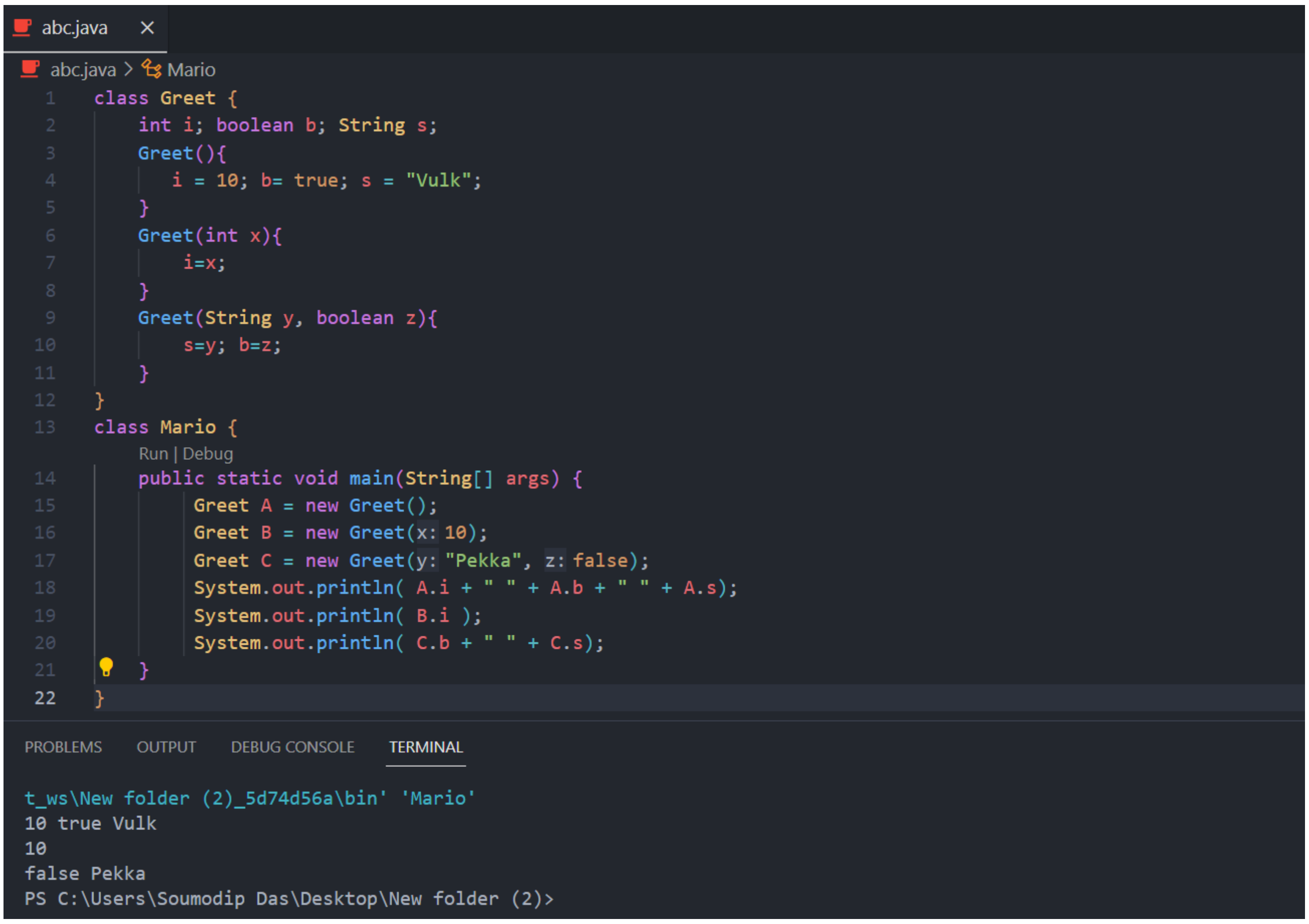
Constructor Overloading :-

What is Constructor Overloading?

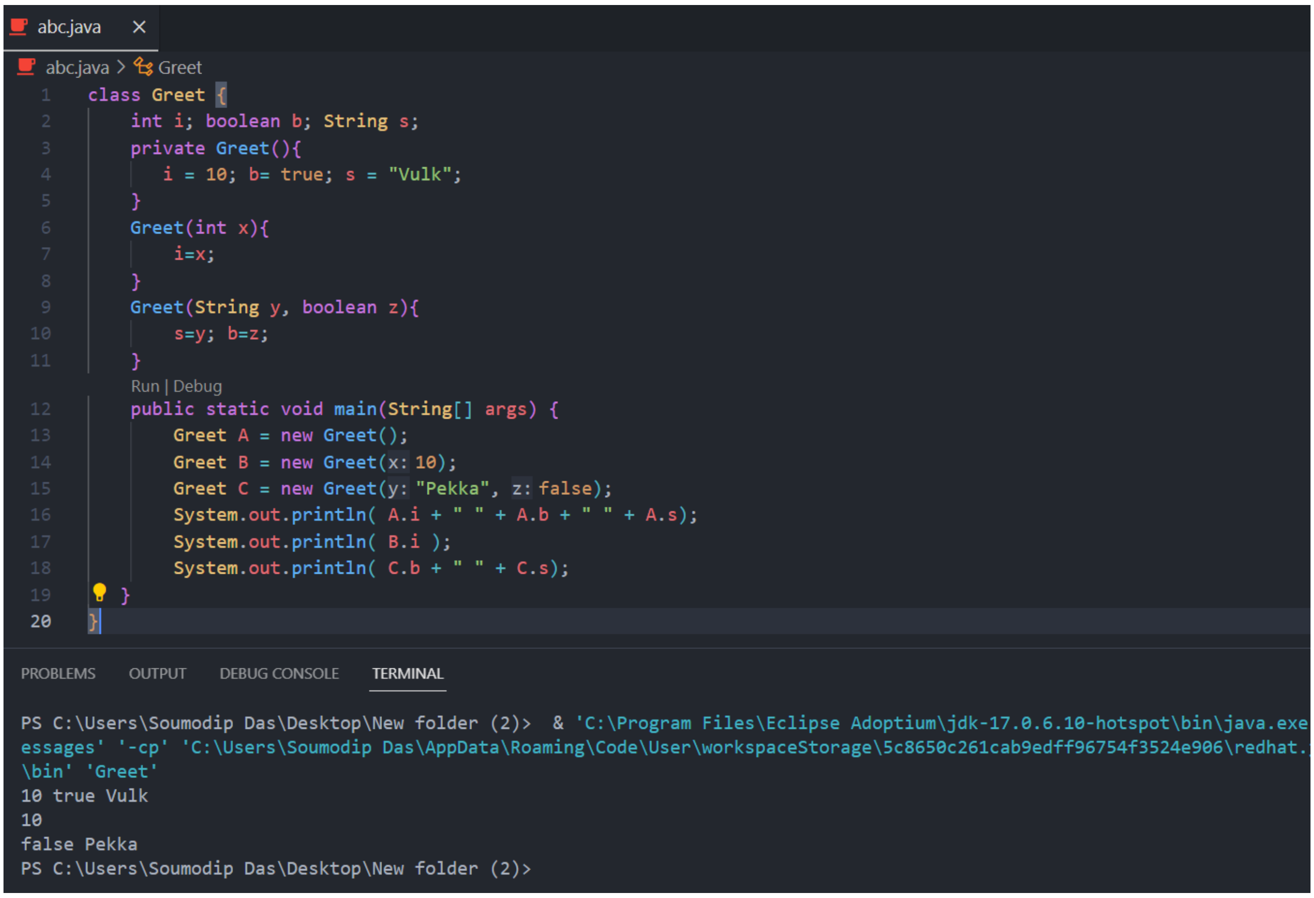
Whenever we have more than one constructor in our class then it is known as constructor overloading.



```java
class Greet {
    int i; boolean b; String s;
    Greet(){
        i = 10; b= true; s = "Vulk";
    }
    Greet(int x){
        i=x;
    }
    Greet(String y, boolean z){
        s=y; b=z;
    }
}
class Mario {
    public static void main(String[] args) {
        Greet A = new Greet();
        Greet B = new Greet(x: 10);
        Greet C = new Greet(y: "Pekka", z: false);
        System.out.println( A.i + " " + A.b + " " + A.s);
        System.out.println( B.i );
        System.out.println( C.b + " " + C.s);
    }
}
```

```
t_ws\New folder (2)_5d74d56a\bin' 'Mario'
10 true Vulk
10
false Pekka
PS C:\Users\Soumodip Das\Desktop\New folder (2)>
```

Using a private constructor :-



abc.java

abc.java > Greet

```java
class Greet {
    int i; boolean b; String s;
    private Greet(){
        i = 10; b= true; s = "Vulk";
    }
    Greet(int x){
        i=x;
    }
    Greet(String y, boolean z){
        s=y; b=z;
    }
    Run | Debug
    public static void main(String[] args) {
        Greet A = new Greet();
        Greet B = new Greet(x: 10);
        Greet C = new Greet(y: "Pekka", z: false);
        System.out.println( A.i + " " + A.b + " " + A.s);
        System.out.println( B.i );
        System.out.println( C.b + " " + C.s);
    }
}
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
PS C:\Users\Soumodip Das\Desktop\New folder (2)>  & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.6.10-hotspot\bin\java.exe
essages' '-cp' 'C:\Users\Soumodip Das\AppData\Roaming\Code\User\workspaceStorage\5c8650c261cab9edff96754f3524e906\redhat.
\bin' 'Greet'
10 true Vulk
10
false Pekka
PS C:\Users\Soumodip Das\Desktop\New folder (2)>
```

Static Block :-

What is static block?

In java, it is a such kind of block which gets executed at the time of loading . class file into JVM memory.

```
Class name {
    Static {
    }
}
```

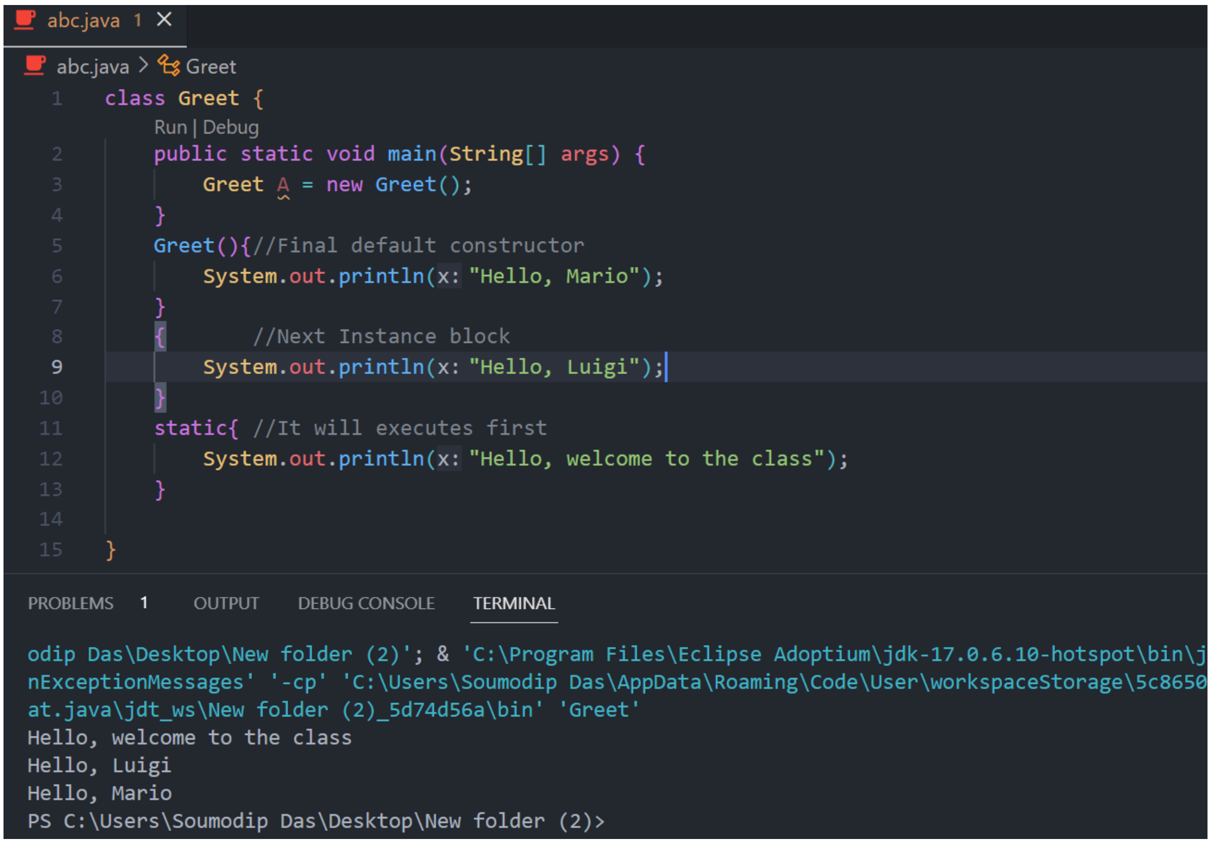
Instance Block :-

What is instance block?

It is same as method which has no name, it can be written inside a class only but not inside the method.

- It always executed before the constructor.
- We can use variable only inside the instance block not method.
- We write time consuming code inside an instance block like – JDBC.

```
Class name {
        {
        }
}
```



abc.java 1 ×

abc.java > Greet

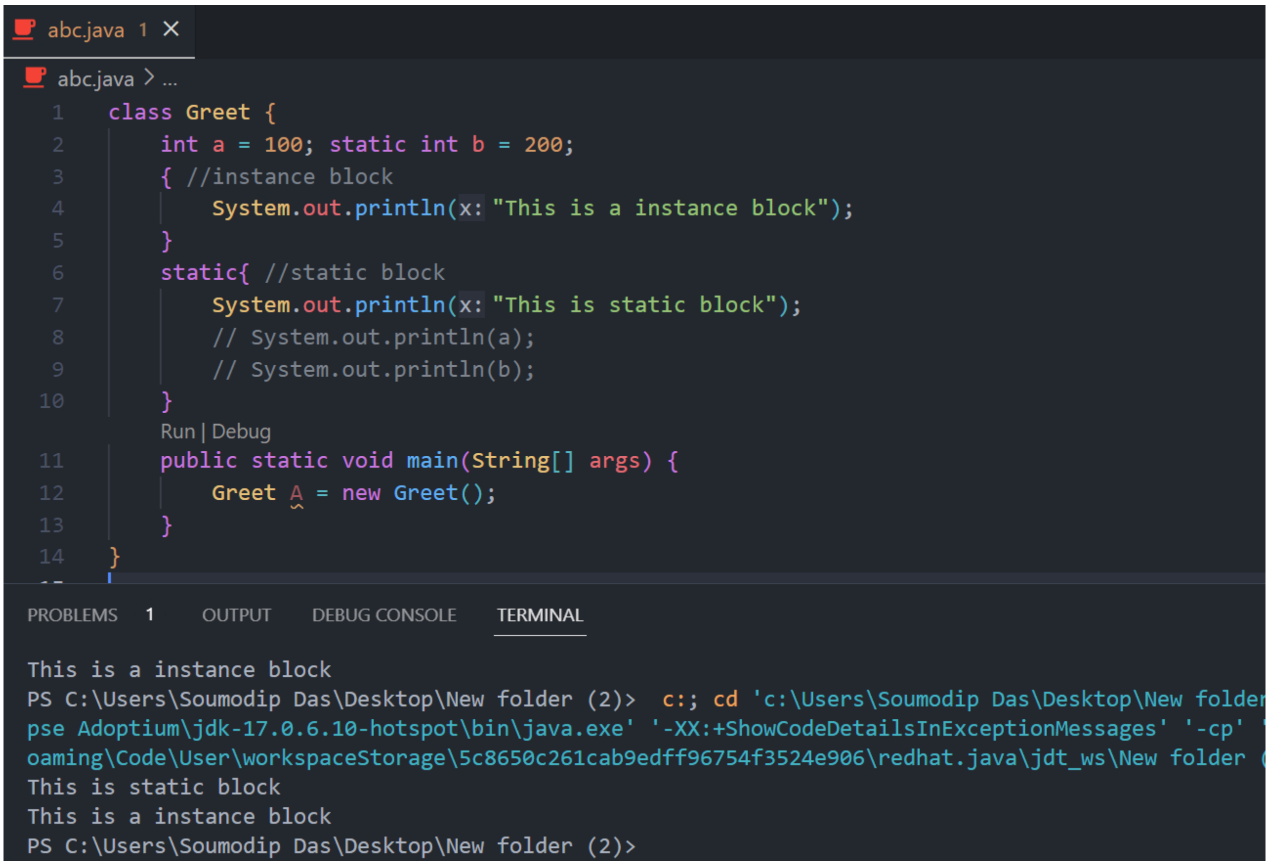
```java
class Greet {
    Run | Debug
    public static void main(String[] args) {
        Greet A = new Greet();
    }
    Greet(){//Final default constructor
        System.out.println(x: "Hello, Mario");
    }
    {       //Next Instance block
        System.out.println(x: "Hello, Luigi");
    }
    static{ //It will executes first
        System.out.println(x: "Hello, welcome to the class");
    }

}
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
odip Das\Desktop\New folder (2)'; & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.6.10-hotspot\bin\j
nExceptionMessages' '-cp' 'C:\Users\Soumodip Das\AppData\Roaming\Code\User\workspaceStorage\5c8650
at.java\jdt_ws\New folder (2)_5d74d56a\bin' 'Greet'
Hello, welcome to the class
Hello, Luigi
Hello, Mario
PS C:\Users\Soumodip Das\Desktop\New folder (2)>
```

Instance block vs static block :-



```java
class Greet {
    int a = 100; static int b = 200;
    { //instance block
        System.out.println(x: "This is a instance block");
    }
    static{ //static block
        System.out.println(x: "This is static block");
        // System.out.println(a);
        // System.out.println(b);
    }
    public static void main(String[] args) {
        Greet A = new Greet();
    }
}
```

```
This is a instance block
PS C:\Users\Soumodip Das\Desktop\New folder (2)>  c:; cd 'c:\Users\Soumodip Das\Desktop\New folder
pse Adoptium\jdk-17.0.6.10-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' '
oaming\Code\User\workspaceStorage\5c8650c261cab9edff96754f3524e906\redhat.java\jdt_ws\New folder (
This is static block
This is a instance block
PS C:\Users\Soumodip Das\Desktop\New folder (2)>
```

Inheritance :-

What is inheritance?

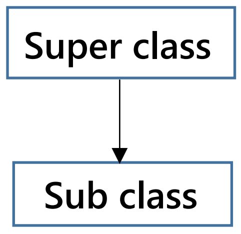
When we construct a new class from existing class is such way that the new class access all the features and properties of existing class called inheritance.

- Extends keyword is used to perform inheritance.
- It provides code reusability.
- We cannot access private members of class through inheritance.
- A sub class contains all the features of super class so, we should create object of sub class.
- Method overriding only possible through inheritance.

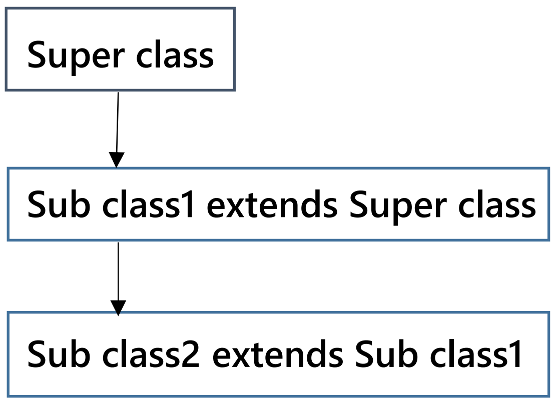
```
Class Greet {
}
Class Mario extends Greet {
}
```

Types :-

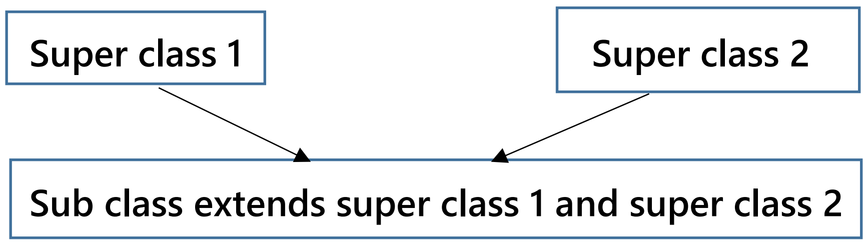
1. Simple/Single inheritance



2. Multilevel inheritance

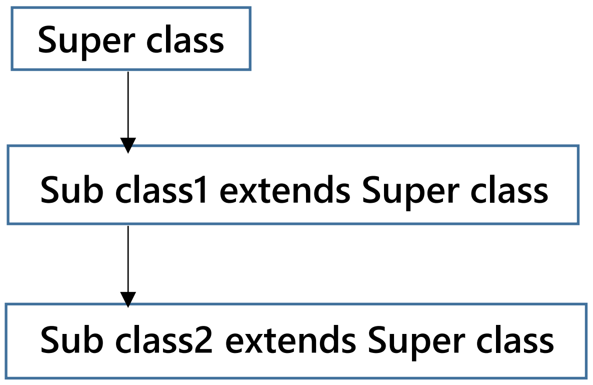


3. Multiple inheritance



Doesn't

4. Hierarchical inheritance


Super class
Sub class1 extends Super class
Sub class2 extends Super class

Simple Inheritance :-

What is simple inheritance?

It is nothing but which contain only one super class and only one sub class is called simple inheritance.

```
Super class {
 }
 Sub class extends Super class {
 }
```

Multilevel Inheritance :-

What is multilevel inheritance?

In it we have only one super class and multiple sub classes is called multilevel inheritance.

```
Super class {
 }
Sub1 class extends Super class {
 }
Sub2 class extends Sub1 class {
 }
```

Multiple Inheritance :-

Java does not multiple inheritance

Why java does not support multiple inheritance?

Whenever a sub class wants to inherit the property of two or more super classes, that have same method, javac cannot decide which class's method it should inherit.

Then there might be a chance of memory duplication. That's why java does not support multiple inheritance through classes.

```
Class Greet {                    |        Class Mario {
     Method                      |             Method
}                                |        }

        Class Luigi extends Greet, Mario {
              Luigi will be confused
        }
```

Hierarchical Inheritance :-

What is hierarchical inheritance?

An inheritance which contain only one super class and multiple sub classes and all sub classes directly extends super class then it is called hierarchical inheritance.

```
                      Class Greet {


                       }



Class Mario extends Greet {        Class Luigi extends Greet  {


            }                                 }
```