Class & Object

Class (Logical Entity)

- A class is a user defined blueprint or Prototype from where the objects can be created.
- Using (;) at the end of the class is optional

```
class Example{

};
```

Class consists of,

- The class name begins with CapitalLetter
- Body surrounded by {}
    - Data – Called fields
    - Operation – Called methods

A class cannot be either private or protected except nested class

```
public class Customer {
    String Name;
    String Size;

    public Customer(String name, String size) {
        Name = name;
        Size = size;
    }

    public String getName() {
        return Name;
    }

    public void setName(String name) {
        Name = name;
    }

    public String getSize() {
        return Size;
    }

    public void setSize(String size) {
        Size = size;
    }
}
```

Object (Runtime Entity)

- Object is an instance of a class.
    - State
    - Identity
    - Behaviour
- New is the keyword used to allocate memory at Run time.
- All objects get the memory in Heap.

```
ClassObject obj = new ClassObject();
//        obj                 -> Object Reference
//        ClassObject()       -> Instance created
//        new ClassObject()   -> Memory Allocated
```

Memory management

```
class Clothing {
    double total, price;
    String description, size;
    public Clothing(){

    }
    public Clothing(double total, double price, String description, String
size) {
        this.total = total;
        this.price = price;
        this.description = description;
        this.size = size;
    }

    public static void main(String[] args){
        double newPrice = 20;
        Clothing c1 = new Clothing(1.25,15,"Shirt","M");
        Clothing c2 = new Clothing();
        c2.description = "Socks";
    }
}
```

Stack
newPrice = 20 c1 = 0x11 c2 = 0x12

Heap
0x11 total = 1.25 price = 15 description = Shirt size = M
0x12 total = 0.0 price = 0.0 description = Socks size = NULL

```
Note :- Object reference also make change
class BM {
    int a = 10;
    public static void main(String[] args) {
        BM obj = new BM();
        obj.change(obj);
        System.out.println(obj.a);
//      Output = 20
    }
    public void change(BM tempObj) {
        tempObj.a=20;
    }
}
```

- Variable Declaration in Java without the initialization doesn't allocate memory

```java
int a;        // -> Declaration
int Aa = 10; // -> Instantiation
int b;
b = 5;        // -> Initialization
```

Constructor

- It is a special method used to initialize objects
- Methods can have same class name. But they are not constructors.
- No return type for constructor
- Cannot declare a constructor as final
- First the constructed called, when the object of the class created.

Default Constructor

- Default constructor is used to provide default value to the fields/objects (0/NULL)

Parameterized Constructor

- We can use Getters (Accessor) & Setters (Mutator) to utilize the fields.
    - **Accessor** -> Used to read the instance variable
    - **Mutator** -> Modify the variable values

Blocks (In predence)

- Static block -> static {}
- Instance block -> {} -> Whenever object created it runs
- Constructor -> ClassName{} -> Whenever object created it runs
- Method -> run(){}