❖ <u>Class & Object</u>

- A class[logical entity] is a definition block which defines the state & behaviour of the object.
- An entity which has its own states & behaviour is known as object[physical entity].
- The state represents the characteristics of object whereas behaviour represents the action or the functionality of the object.
- The object doesn’t exist without a class definition.
- We can create any number of instances from a class each instance differs in the values of the state but same in behaviour.
- Defining anything in the body of a class is known as members of the class.
- The class body can contain member variable or member functions.
- Member variables are the variables declared & initialize in the body of the class where as the member function are the methods defined in the body of the class.
- The member variable are used to represent the data where as member function are used to represent the operation performed on the data.

❖ <u>Member types</u>

- We can define 2 types of members in a class body.
- static member types
- non – static member types
- The static member types are declared using static keyword it is also known as a class member because it is associated to the class.
- The static members of the class can be accessed in any class body by using the syntax.

  className.memberName
- The static members of a class are loaded one copy in the memory which can be modified.
- The non-static members are defined without static keyword.
- It is associated to the instance of the class, hence it is also known as **Instance member**.
- To access the non-static members of a class we should create the instance of the class.
- The instance is created by using **new operator**.
- Whenever we create an instance of a class, a copy of the class gets loaded multiple copies in the memory.

- We can create ‘n’ no. of instance.
- The non-static members are loaded multiple copies in the memory.
- To access each instance we should create reference variable.
- The reference variable are used to identify the instance & to access the instance variables & instance functions.
- The reference variable are declared by using class type & it is non-primitive variables.
- Changes made to the instance variable of a instance will not reflect in other instances.
- An instance can be referred by any number of reference variables. In such case, if we change the instance property using one reference will be reflects in other references.

❖ <u>NOTE</u>

- A reference variable should always hold the address where the instances created in the memory. If not the reference variable should hold **null** or in other word a reference variable should point to instance if not points to null.

- If a reference variable is pointing to null & if we perform any operation using the reference then **JVM** throws **NullPointerException.**

❖ <u>Constructor</u>

- Constructor is a special member of a class used to provide the initialization to the instance variables of the class at the time of object creation.

- Every class defined in Java language should specified the constructor.

- The constructor can be defined either by compiler or by user.

- Based on who defines, constructor has 2 types

    1. Compiler defined constructor

    2. User defined constructor

1. Compiler defined constructor
    - This will not have any parameter & it is known as default constructor.
    - The compiler defines the constructor only when the class is not having any user defined constructor.
    -
2. User defined constructor
    - The constructor defined by user is known as user defined constructor.

    - It is of 2 types

1. Constructor without argument (No argument constructor)
2. Constructor with argument (Parameterized constructor)

- A constructor defined with parameterized constructor is known as a parameterized constructor.
- While defining a constructor the constructor name must be same has class name.
- The constructor should not specify any return type.

```
class Demo{
        int x;
        Demo(){
                x = 10;
        }
}
```

- If the constructor is defined with a return type then it will be treated as member function.
- If class has blocks & constructors then JVM executes blocks first & then constructor of the class.
- A constructor cannot be declared as **static.**
- In a class we can define any number of constructors provided the argument types of the constructor should vary, such constructors are known as **Overloaded constructors.**
- In other words, defining multiple constructors with different argument list is known as **Constructor Overloading.**
- We cannot define two constructors with same argument types.

❖ <u>**Need for Overloaded Constructor**</u>

- The overloaded constructors will help to create objects with different initialization of instance variable.