iamneo

Day 1

Class and objects / Constructors

Class

A Class is a user-defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type.

Class is a set of object which shares common characteristics/ behavior and common properties/ attributes.

Class is a group of variables of different data types and group of methods.
Class does not occupy memory.

A Class Contain:

- data member
- method
- nested class and
- Constructor

Example for class

```
class Employee{
    int id;
    String name;

    public static void main(String args[])
    {
        Employee s1 = new Employee();
        System.out.println(s1.id);
        System.out.println(s1.name);
    }
}
```

Object

- A Java object is a member (also called an instance) of a Java class. Each object has an identity, a behavior and a state.

- The state of an object is stored in fields (variables), while methods (functions) display the object's behavior.

- **State**: It is represented by attributes of an object. It also reflects the properties of an object.
- **Behavior**: It is represented by the methods of an object. It also reflects the response of an object with other objects.
- **Identity**: It gives a unique name to an object and enables one object to interact with other objects.

- In **Java**, we cannot execute any program without creating an **object**

How to Create Object in Java

- The **new** keyword in Java instantiates a class by allocating desired memory for an associated new object.

- When we create an instance of the class by using the new keyword, it allocates memory (heap) for the newly created **object** .

- Using the obj variable, we can access the members of the new object as shown in the output.

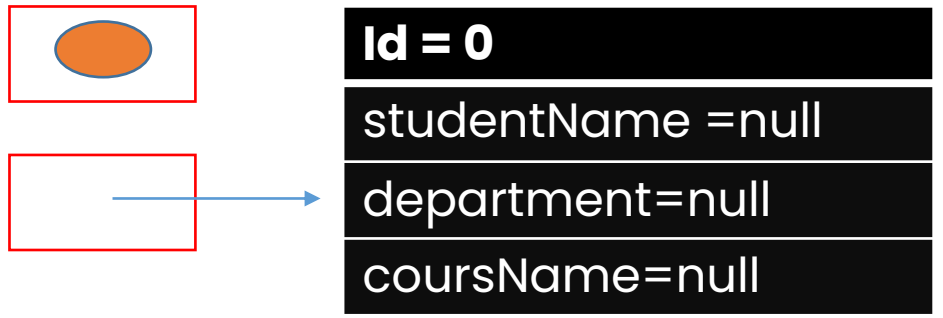
Syntax

```
Student  stuobj  = new  Student();   // constructor.
```

- The new keyword returns a reference to that memory post object creation.
- The new keyword allocates memory to the new objects at runtime.

Example

Statement	Effect
Student stuobj	stuobj
Stuobj =new Student	stuobj



Constructors In Java

- Constructor is a special type of method that is used to initialize objects of a class.
- It is called when an instance (object) of the class is created using the "new" keyword. The constructor has the same name as the class and it does not have a return type, not even void.
- The primary purpose of a constructor is to set the initial values of instance variables or properties of the object being created.
- Constructors can also perform additional tasks such as allocating memory or establishing connections to external resources.

Rules for Creating Constructor

- Constructor name must be the same as its class name
- A Constructor must have no explicit return type
- A Java constructor cannot be abstract, static, final, and synchronized

Types of Constructor

There are two types of constructors in Java

- Default constructor (no-arg constructor)
- Parameterized constructor

Default Constructor

- A default constructor in Java is a special type of constructor that is automatically created by the Java compiler if a class does not have any constructors explicitly defined.
- A default constructor has no parameters, and its body is empty. It is also called a no-argument constructor or zero-argument constructor.
- The primary purpose of a default constructor is to provide a default initialization of the instance variables of a class.
- The default constructor is used to provide the default values to the object like 0, null, depending on the type.

Code

```
class Employee{
 int id;
 String name;
Employee()
{
System.out.println(id+" " + name );
}
public static void main(String args[]){
Employee b=new Employee();
}
}
```

Output:

0 null

The default constructor provides 0 and null values by default for the instance variables of a class.

Parameterized Constructor

- A parameterized constructor is a type of constructor in Java that takes a specific number of parameters.

- The purpose of using a parameterized constructor is to allow objects of the same class to be initialized with different values for their instance variables.

- While it is possible to use the same values for different objects, the primary use of a parameterized constructor is to provide custom initialization for each object.

code:

```
class Employee{
    int id;
   String name;
    int age;
    Employee(int i,String n){
    id = i;
    name = n;
    }
    Employee(int i,String n,int a){
    id = i;
    name = n;
    age=a;
    }
    void display(){System.out.println(id+" "+name+" "+age);}
    public static void main(String args[]){
    Employee  emp1 = new Employee(100,“Stark");
    Employee  emp2 = new Employee(201,“Elon Musk",25);
    emp1.display();
    emp2.display();
   }
}
```

Difference between Constructor and Method

Constructor	Method
A Constructor is a block of code that initializes a newly created object.	A Method is a collection of statements which returns a value upon its execution.
A Constructor can be used to initialize an object.	A Method consists of Java code to be executed.
A Constructor is invoked implicitly by the system.	A Method is invoked by the programmer.
A Constructor is invoked when a object is created using the keyword **new**.	A Method is invoked through method calls.
A Constructor doesn't have a return type.	A Method must have a return type.
A Constructor's name must be same as the name of the class.	A Method's name can be anything.