**Course 2 - Backend and Database Development**

**Day 3: 12 Dec 2024**

**Constructor :**

Constructor is a type of special method. which help to create the memory or object.

While creating or writing constructor in user defined class we need to remember few points.

1. Constructor have same name as class itself.
2. Constructor doesn’t contain return type not even void also.
3. Constructor no need to call it will get call automatically when we create the object.
4. If we doesn’t write any constructor by default. Default constructor provided by JVM(Java Virtual machine). Default constructor always empty constructor.
5. We can write empty as well as parameter constructor. Constructor mainly use to do some initialization.

Encapsulation ; Binding or wrapping data(variables) and code(methods or functions) in a single unit is known as Encapsulation.

Example : class.

If class contains any variable and method we can’t access those variable and method without creating object of that class.

Inheritance : Inheritance is use to inherits properties and behaviour of old class to new class. Inheritance mainly use to do re-usability.

class OldClass{ // super class or base class or parent class

properties or variables

behaviour or method

}

class NewClass extends OldClass { // sub class or derived class or child class

properties or variables

behaviour or method

}

OldClass obj1 = new OldClass();

obj1.oldClassvariable=10;

obj1.oldClassMethod();

NewClass obj2 = new NewClass();

obj2.newClassvariable=20;

obj2.newClassMethod();

obj2.oldClassVariable=30;

obj2. oldClassMethod();

Types of Inheritance

1. Single Inheritance : one super class and one sub class.

class A { }

class B extends A{ }

1. Multilevel inheritance : one super and n number of sub classes connected one by one

class A { }

class B extends A {}

class C extends B{}

class D extends C{}

1. Hierarchical inheritance : one super class and n number of class classes connected directly to super class.

class A { }

class B extends A {}

class C extends A{}

class D extends A{}

1. Multiple inheritance : more than one super class and one sub class

class A {}

class B {}

class C extends A,B{} :Error : Because Java doesn’t support multiple inheritance. This type of inheritance in Java we can achieve using interface.

OOPs relationship

Manager, Developer is a Employee

has a relationship

Employee/Manager has a Address

class Employee {

id,name,salary

readEmp, disEmp

}

class Manager extends Employee{

numberOfEmp

Address add = new Address();

readMgr,disMgr

}

class Developer extends Employee{

projectName;

readDev,disDev

}

class Address {

city,state

readAdd,disAdd

}

Polymorphism

One name many forms or many implementations.

Types of polymorphism

1. Compile time polymorphism or static binding or early binding

Example : Method Overloading :

The method have same name but different parameter list(type of parameter list or number of parameter list must be different).

class Operation {

read() : we can do the addition of number

dis(): we can do the addition of number

add(): we can do the addition of number

}

1. Run time polymorphism or dynamic binding or late binding

Example : Method Overriding :

: The method have same name and same method signature( number of parameter list, type of parameter list and return type must be same).

To achieve method overriding we need inheritance.

class Bike {

void speed() {

System.out.println(“60km/hr”)

}

}

class Pulsar extends Bike{

void speed() {

System.out.println(“90km/hr”)

}

void color() {

System.out.println(“Black”);

}

}

class Honda extends Bike{

void color() {

System.out.println(“Gray”);

}

}