Day 5: Oops

**Java Bean class**

In JavaBean class all variable must be private

For each variable we need to provide setter and getter methods.

Setter to set the value and getter to get the value.

Pure encapsulation class.

**Inheritance :**

Inheritance is use to inherits properties and behaviour of old class to new class.

class OldClassName { super class, base class or parent class

property

behaviour

} sub class, child class or derived class.

class NewClassName extends OldClassName{

property

behaviour

}

Type of inheritance

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

class A {}

class B extends A{}

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

class A {}

class B extends A{}

class C extends B{}

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

class A {}

class B extends A{}

class C extends A{}

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

Class A {}

Class B {}

Class C extends A,B{} error : Java doesn’t supper multiple inheritance it support using interface concept.

Oops relationship

Manager is a Employee

Employee has a Address

class Employee {

Id,name,salary (private)

Scanner obj = new Scanner(System.in);

Address add =new Address();

readEmployee() {

enter the id, name, salary

add.readAdd();

}

disEmployee(){

display id,name,salary

add.disAdd();

}

}

class Manager extends Employee{

numberOfEmp;

readManager() {

numberOfEmp value need receive

}

disManager() {

display number of employee

}

}

class Developer extends Employee{

projectName;

readDev() {

read projectName

}

disDev() {

dis projectnames

}

}

Class ProjectManager extends Manager{

clientName

readPmgr() {

read client details

}

disPmgr() {

display client details

}

}

class Address {

city, state

Scanner obj = new Scanner()

readAdd() {} read city and state

disAdd() {} display city and state

}

**Polymorphism :** one name many forms or many implementations

2 types

1. Compile time polymorphism or static binding or early binding

Method overloading :

Method have same name and different parameter list(type of parameter or number of parameter).

1. Run time polymorphism or late binding or dynamic binding

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