Inheritance:

Process of deriving a new class from existing class

**Properties:**

* It can access base class methods and variables
* It can have own functions
* It can override the base class functions
* It follows is-relationship
* Base class constructor also called first during subclass object creation

**Type:**

->single

->multilevel

->multiple

->hierarchy

->Hybrid





Multiple -> cant be implemented in inheritance

**Syntax:**

Class subclassname extends baseclassname{

//variables

//Methods

}

Eg:

class A{

int a;

void display(){

a=10;

System.out.println("----- Printing a value inside Baseclass display()-----");

System.out.println("a="+a);

}

}

class B extends A{

int b;

void print(){

a=100;

b=20;

System.out.println("----- Printing a and b values inside subclass print()-----");

System.out.println("a="+a+"\nb="+b);

}

}

class HelloWorld {

public static void main(String[] args) {

B obj = new B();

obj.display();

obj.print();

A a1 = new A();

}

}

Output:

----- Printing a value inside Baseclass display()-----

a=10

----- Printing a and b values inside subclass print()-----

a=100

b=20

**Base class constructor also called first during subclass object creation**

// Online Java Compiler

// Use this editor to write, compile and run your Java code online

class A{

int a;

A(){

System.out.println("--- A class Constuctor called ---");

}

void display(){

a=10;

System.out.println("----- Printing a value inside Baseclass display()-----");

System.out.println("a="+a);

}

}

class B extends A{

B(){

System.out.println("--- B class Constuctor called ---");

}

int b;

void print(){

a=100;

b=20;

System.out.println("----- Printing a and b values inside subclass print()-----");

System.out.println("a="+a+"\nb="+b);

}

}

class HelloWorld {

public static void main(String[] args) {

B obj = new B();

}

}

Output:

--- A class Constuctor called ---

--- B class Constuctor called ---

**The base class variables values are initialized through their constructors only**

class A{

int a;

A(int a){

this.a=a;

System.out.println("--- A class Constuctor called ---");

}

void display(){

System.out.println("----- Printing a value inside Baseclass display()-----");

System.out.println("a="+a);

}

}

class B extends A{

int b;

B(int a,int b){

super(a);

this.b = b;

System.out.println("--- B class Constuctor called ---");

}

void print(){

System.out.println("----- Printing a and b values inside subclass print()-----");

System.out.println("a="+a+"\nb="+b);

}

}

class HelloWorld {

public static void main(String[] args) {

B obj = new B(10,20);

obj.print();

}

}

Output:

--- A class Constuctor called ---

--- B class Constuctor called ---

----- Printing a and b values inside subclass print()-----

a=10

b=20

**Multilevel Inheritance**

->it can access grandfather, father classes

Eg:

class Animal{

void eat()

{

System.out.println("eating...");

}

}

class Dog extends Animal{

void bark()

{

System.out.println("barking...");

}

}

class BabyDog extends Dog

{

void nature()

{

System.out.println("weeping...");

}

}

class TestInheritance2{

public static void main(String args[]){

BabyDog d=new BabyDog();

d.nature();

d.bark();

d.eat();

}

}

**Accessing the the base class method in subclass method**

**Super -> used**

class Animal{

void eat()

{

System.out.println("eating...");

}

}

class Dog extends Animal{

void bark()

{

super.eat();

System.out.println("barking...");

}

}

class BabyDog extends Dog

{

void nature()

{

super.bark();

System.out.println("weeping...");

}

}

class TestInheritance2{

public static void main(String args[]){

BabyDog d=new BabyDog();

d.nature();

}}

**o/p:**

eating...

barking...

weeping...

**assigning values of classes variables using its constructor only in multilevel**

class A{

int a;

A(int a){

this.a=a;

System.out.println("--- A class Constuctor called ---");

}

void display(){

System.out.println("----- Printing a value inside Baseclass A of display()-----");

System.out.println("a="+a);

}

}

class B extends A{

int b;

B(int a,int b){

super(a);

this.b = b;

System.out.println("--- B class Constuctor called ---");

}

void print(){

System.out.println("----- Printing a and b values inside B Class print()-----");

System.out.println("a="+a+"\nb="+b);

}

}

class C extends B{

int c;

C(int a,int b,int c){

super(a,b);

this.c=c;

System.out.println("--- C class Constuctor called ---");

}

void show(){

System.out.println("----- Printing a,b,c values inside C class show()-----");

System.out.println("a="+a+"\nb="+b+"\nc="+c);

}

}

class HelloWorld {

public static void main(String[] args) {

C obj = new C(10,20,30);

obj.show();

}

}

**o/p:**

--- A class Constuctor called ---

--- B class Constuctor called ---

--- A class Constuctor called ---

----- Printing a,b,c values inside C class show()-----

a=10

b=20

c=30

-------------- hierarchy -----------------------

class Animal{

void eat()

{

System.out.println("eating...");

}

}

class Dog extends Animal{

void nature()

{

System.out.println("barking...");

}

}

class Cat extends Animal

{

void nature()

{

System.out.println("Meow...");

}

}

class TestInheritance2{

public static void main(String args[]){

Dog d=new Dog();

Cat c= new Cat();

d.nature();

c.nature();

}

}

o/p:

barking...

Meow...

final keyword:

used for prevent a class from inheritance

final class A{

void display(){

System.out.println("i m display() in A");

}

}

class B extends A{

void show(){

System.out.println("i m show() in B");

}

}

class TestInheritance2{

public static void main(String args[]){

B b1=new B();

b1.display();

}

}

o/p:

ERROR!

javac /tmp/pizaozexNj/TestInheritance2.java

/tmp/pizaozexNj/TestInheritance2.java:6: error: cannot inherit from final A

class B extends A{

^

1 error

**task:**

**single**

1.Define a Circle class with areaofcircle() method and radius, area\_circle are to double variable. Also define subclass Sphere from Circle class will having areaofsphere() and area\_Sphere(double) variable. calculate area of circle and area of sphere using the subclass objects only

Note:

area of circle -> π r2

area of sphere  -> 4 π r2

2

extends circle class to the Perimeter class having Perimeter variable(double) and function area\_perimeter can use of radius value from circle base class to calculate perimeter of circle . Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square.

**multi level:**

Create a base class Employee having variables Name, ID,Department,Address, DOB. Its constructor is used to initialize these variables. Let Assume a class Operator cextends from Employee having double attendance, perday\_salary, monthly\_salary,Overtime,Overtimepay, String , String message. boolean eligibilitystatus and BonusCalculator () function for validating employee is eligible for bonus using following condition,

attendance is greater than or equal to 28

overtime is greater than or equal to 50

and Its constructor is used to initialize these variables. Let Assume a class SalaryCalculator extends from Operator having variables bonus and monthlySalaryCalculator () Function calls the BonusCalculator() of Operator class and to calculate salary based on the eligibilitystatus variable value.if they eligible then bonus will be added to the monthly\_salary otherwise without bonus salary will be paid.

**Hierarchy**

Create Circle base class and define variable radius(double) and initialize the variable through constructor.

and extends circle class to the area class having area variable(double) and function area\_circle can use of radius value from circle base class to calculate area of circle

ans:

single:

1.define a Circle class with areaofcircle() method and radius, area\_circle are to double variable. Also define subclass Sphere from Circle class will having areaofsphere() and area\_Sphere(double) variable. calculate area of circle and area of sphere using the subclass objects only

Note:

area of circle -> π r2

area of sphere  -> 4 π r2

import java.util.Scanner;

class Circle{

double radius;

double areaofcircle;

Circle(double radius){

this.radius = radius;

}

void area\_circle(){

areaofcircle=3.14\*radius\*radius;

System.out.println("The area of the circle is "+areaofcircle);

}

}

class Sphere extends Circle {

double areaofsphere;

Sphere(double radius){

super(radius);

}

void area\_sphere(){

double areaofsphere;

areaofsphere=4\*3.14\*radius\*radius;

System.out.println("The area of the Sphere is "+areaofsphere);

}

}

class mainclass{

public static void main(String args[]){

Scanner sc= new Scanner(System.in);

System.out.print("Enter the radius:");

double radius=sc.nextDouble();

Sphere s=new Sphere(radius);

s.area\_circle();

s.area\_sphere();

}

}

o/p:

Enter the radius:1.1

The area of the circle is 3.7994000000000008

The area of the Sphere is 15.197600000000003

**multi level:**

Create a base class Employee having variables Name, ID,Department,Address, DOB. Its constructor is used to initialize these variables. Let Assume a class Operator cextends from Employee having double attendance, perday\_salary, monthly\_salary,Overtime,Overtimepay, String message. boolean eligibilitystatus and BonusCalculator () function for validating employee is eligible for bonus using following condition,

attendance is greater than or equal to 28

overtime is greater than or equal to 50

and Its constructor is used to initialize these variables. Let Assume a class SalaryCalculator extends from Operator having variables bonus and monthlySalaryCalculator () Function calls the BonusCalculator() of Operator class and to calculate salary based on the eligibilitystatus variable value. if they eligible then bonus will be added to the monthly\_salary otherwise without bonus salary will be paid.

import java.util.Scanner;

class Employee{

String Name,Address,DOB,ID,Department;

Employee(String Name,String ID, String DOB, String Department,String Address){

this.Name=Name;

this.ID = ID;

this.DOB= DOB;

this.Department = Department;

this.Address=Address;

System.out.println("Welcome "+ Name);

}

}

class Operator extends Employee{

double attendance, perday\_salary, monthly\_salary,Overtime,Overtimepay;

String message;

boolean eligibilitystatus;

Operator(String Name,String ID,String DOB,String Department,String Address,double attendance,double perday\_salary,double Overtime,double Overtimepay){

super(Name,ID,DOB,Department,Address);

this.attendance=attendance;

this.perday\_salary=perday\_salary;

this.Overtime=Overtime;

this.Overtimepay=Overtimepay;

}

void BonusCalculator(){

if(attendance>=28 ){

if(Overtime>=50){

eligibilitystatus=true;

message = "Eligible for bonus";

}

else{

eligibilitystatus=false;

message = "Not eligible for bonus Bcz less Overtime";

}

}

else{

eligibilitystatus=false;

message = "Not eligible for bonus Bcz less Attendance";

}

}

}

class SalaryCalculator extends Operator{

double Bonus;

SalaryCalculator(String Name,String ID,String DOB,String Department,String Address,double attendance,double peray\_salary,double Overtime,double Overtimepay ){

super(Name,ID,DOB,Department,Address,attendance,peray\_salary,Overtime,Overtimepay);

}

void monthlySalaryCalculator(){

Scanner sc= new Scanner(System.in);

System.out.println("Enter the bonus Amount for this month:");

Bonus = sc.nextDouble();

BonusCalculator();

if(eligibilitystatus == true){

monthly\_salary =(attendance \* perday\_salary) + (Overtime \* Overtimepay) + (Bonus \* Overtime);

System.out.println("Bonus="+(Bonus\*Overtime));

System.out.println("Your Monthly salary = "+ monthly\_salary);

}

else{

System.out.println(message);

monthly\_salary =(attendance\*perday\_salary) + (Overtime \* Overtimepay);

System.out.println("Your Monthly salary = "+monthly\_salary);

}

}

}

class Mainclass {

public static void main(String[] args) {

System.out.println(" --- Enter the New employee details ----");

Scanner sc= new Scanner(System.in);

System.out.print("Enter the Name:");

String Name = sc.next();

System.out.print("Enter the ID:");

String ID = sc.next();

System.out.print("Enter the DOB:");

String DOB = sc.next();

System.out.print("Enter the Department:");

String Deparment = sc.next();

System.out.print("Enter the Address :");

String Address = sc.next();

System.out.println(" --- Enter details for salary calculator----");

System.out.print("Enter the attendance:");

double attendance = sc.nextDouble();

System.out.print("Enter the perday salary:");

double perday\_salary = sc.nextDouble();

System.out.print("Enter the OverTime:");

double OverTime = sc.nextDouble();

System.out.print("Enter the OverTime pay:");

double Overtimepay = sc.nextDouble();

SalaryCalculator s1 = new SalaryCalculator(Name,ID,DOB,Deparment,Address,attendance,perday\_salary,OverTime,Overtimepay);

s1.nonthlySalaryCalculator();

}

}

o/p:

with bonus:

--- Enter the New employee details ----

Enter the Name:valar

Enter the ID:01

Enter the DOB:2023/1/1

Enter the Department:production

Enter the Address :salem

--- Enter details for salary calculator----

Enter the attendance:30.0

Enter the perday salary:1000.0

Enter the OverTime:50.0

Enter the OverTime pay:10.0

Welcome valar

Enter the bonus Amount for this month:

10.0

Bonus=500.0

Your Monthly salary = 31000.0

--withoutbonus ---

--- Enter the New employee details ----

Enter the Name:valar

Enter the ID:01

Enter the DOB:2023/1/1

Enter the Department:production

Enter the Address :salem

--- Enter details for salary calculator----

Enter the attendance:30.0

Enter the perday salary:1000.0

Enter the OverTime:20

Enter the OverTime pay:100.0

Welcome valar

Enter the bonus Amount for this month:

10.0

Not eligible for bonus Bcz less Overtime

Your Monthly salary = 32000.0

--- withoutbonus ---

--- Enter the New employee details ----Enter the Name:valar

Enter the ID:01

Enter the DOB:2022/1/1

Enter the Department:production

Enter the Address :salem

--- Enter details for salary calculator----

Enter the attendance:25

Enter the perday salary:1000.0

Enter the OverTime:50

Enter the OverTime pay:100.0

Welcome valar

Enter the bonus Amount for this month:

10.0

Not eligible for bonus Bcz less Attendance

Your Monthly salary = 30000.0