ASSIGNMENT 3

1. Create a class Publication with data members title(String) and price(int). From this class derive two classes Book and CD. Class Book adds pages(int) and CD adds Size(int). Each of these classes should have constructors and display(). Write a java program to implement this using super, this and method overriding concepts.

PROBLEM :

class Publication{

int price;

String tittle;

Publication(int price,String tittle){

this.price=price;

this.tittle=tittle;

}

}

class book extends Publication{

int pages;

book(int price,String tittle,int pages){

super(price,tittle);

this.pages=pages;

}

void display(){System.out.println("price of book:"+price+",Tittle of book is: "+tittle+",No of pages in book "+pages);}

}

class CD extends book{

int size;

CD(int price,String tittle,int pages,int size){

super(price,tittle,pages);

this.size=size;

}

void display(){System.out.println("price of book:"+price+",Tittle of book is: "+tittle+",No of pages in book "+pages+"size of book is:"+size);}

}

class TestSuper5{

public static void main(String[] args){

book e1=new book(350,"BATMAN",450);

CD e2=new CD(400,"MARVEL",450,6);

e1.display();

e2.display();

}}

OUTPUT :

price of book:350,Tittle of book is: BATMAN,No of pages in book 450

price of book:400,Tittle of book is: MARVEL,No of pages in book 450size of book is:6

1. Write a simple java program to demonstrate method overriding.

PROBLEM :

class Bank{

int getRateOfInterest(){return 0;}

}

class SBI extends Bank{

int getRateOfInterest(){return 8;}

}

class ICICI extends Bank{

int getRateOfInterest(){return 7;}

}

class AXIS extends Bank{

int getRateOfInterest(){return 9;}

}

class MOVER{

public static void main(String args[]){

SBI s=new SBI();

ICICI i=new ICICI();

AXIS a=new AXIS();

System.out.println("SBI Rate of Interest: "+s.getRateOfInterest());

System.out.println("ICICI Rate of Interest: "+i.getRateOfInterest());

System.out.println("AXIS Rate of Interest: "+a.getRateOfInterest());

}

}

OUTPUT :

SBI Rate of Interest: 8

ICICI Rate of Interest: 7

AXIS Rate of Interest: 9

1. Write a java program to create an interface called Shape with CalculateArea(). Create three classes namely Square,Circle,Triangle which implements Shape.

PROBLEM :

interface Drawable{

void draw();

}

class Square implements Drawable{

public void draw(){System.out.println("drawing Square");}

}

class Circle implements Drawable{

public void draw(){System.out.println("drawing circle");}

}

class Triangle implements Drawable{

public void draw(){System.out.println("drawing triangle");}

}

class TestInterface1{

public static void main(String args[]){

Drawable d=new Circle();

d.draw();

Drawable s=new Square();

s.draw();

Drawable t=new Triangle();

t.draw();

}}

OUTPUT :

drawing circle

drawing Square

drawing triangle

1. Create two packages p1 and p2. The package p1 contains class A which contains one display(). Create class B in package p2. The main method of class B invoke A’s display(). Write a java program to do this.

PROBLEM :

package p1;

public class A{

public void display(){System.out.println("Hello world");}

}

package p2;

import p1.\*;

class B{

public static void main(String args[]){

A obj = new A();

obj.display();

}

OUTPUT :

Hello

1. What is Inheritance?

**Inheritance :** Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

1. What is Multiple Inheritance?

**Multiple Inheritance :**  Object Oriented Programming provides a user the feature of multiple inheritance, wherein a class can inherit the properties of more than a single parent class. In simpler terms, multiple inheritance means a class extending more than one class.

1. What is the use of Super keyword?

**Super keyword :** The super keyword refers to superclass (parent) objects. It is used to call superclass methods, and to access the superclass constructor. The most common use of the super keyword is to eliminate the confusion between superclasses and subclasses that have methods with the same name.

1. What is abstract method?

**Abstract method :** A method without body (no implementation) is known as abstract method. A method must always be declared in an abstract class, or in other words you can say that if a class has an abstract method, it should be declared abstract as well.

1. What is abstract class?

**Abstract class :** A class that is declared using “abstract” keyword is known as abstract class. It can have abstract methods as well as concrete methods. An abstract class cannot be instantiated, which means you are not allowed to create an object of it.

1. What is the use of final modifier?

**Use of final modifier :** The main purpose of using a class being declared as final is to prevent the class from being subclassed. If a class is marked as final then no class can inherit any feature from the final class.

1. What is interface? Write the syntax interface.

**Interface :** An**interface in Java** is a blueprint of a class. It has static constants and abstract methods.

**SYNTAX :** interface <interface\_name>{

// declare constant fields

// declare methods that abstract

// by default.

}

1. What is package?

**package** : A package is a namespace that organizes a set of related classes and interfaces.

1. What is exception?

**Exception :** An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

1. What is the use of finally block?

**Use of finally block** : a block that is used to execute important code such as closing connection, stream etc.