Write a program in Java to demonstrate the uses of classes, objects, and the object-oriented pillars in Java

Program:

//classes and objects

**public** **class** FirstClassProgram {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Makeup ref=**new** Makeup();

ref.creams=2;

ref.nailpolish=5;

ref.face();

**new** Makeup().lips();

}

}

**class** Makeup {

**int** creams,nailpolish,lipsticks;

**void** face() {

System.***out***.println("apply cream on you face to get better skintone:"+creams);

}

**void** lips() {

System.***out***.println("use multicolors");

}

}

Output:

apply cream on you face to get better skintone:2

use multicolors

//Polymorphism

**public** **class** Polymorphism {

**public** **int** sum(**int** x, **int** y)

{

**return** (x + y);

}

**public** **int** sum(**int** x, **int** y, **int** z)

{

**return** (x + y + z);

}

**public** **double** sum(**double** x, **double** y)

{

**return** (x + y);

}

**public** **static** **void** main(String args[])

{

Polymorphism s = **new** Polymorphism();

System.***out***.println(s.sum(5, 20));

System.***out***.println(s.sum(10, 15, 30));

System.***out***.println(s.sum(10.5, 20.5));

}

}

Output:

25

55

31.0

//Inheritance

**package** com.mphasis.training.learning;

**public** **class** InhertitanceDemo {

**public** **static** **void** main(String[] args) {

Aishwarya aishwarya = **new** Aishwarya();

aishwarya.eat();

aishwarya.breathe();

aishwarya.run();

System.***out***.println("-------------------------------");

Akarsh akarsh = **new** Akarsh();

akarsh.eat();

akarsh.breathe();

akarsh.run();

akarsh.swim();

Bindu bindu = **new** Bindu();

// bindu.eat();//Can't call eat from bindu

}

}

**class** Bindu{

}

**class** Human{

**void** eat() {

System.***out***.println("Generic way of eating...");

}

**final** **void** breathe() {

System.***out***.println("Generic way of how Humans breathe...");

}

**void** run() {

System.***out***.println("Generic way of running...");

}

}

**class** Aishwarya **extends** Human{

//@Override

**void** run() {

System.***out***.println("Aishwarya's own way of running...");

}

}

**class** Akarsh **extends** Human{

@Override

**void** eat() {

System.***out***.println("Akarsh's own way of eating...");

}

@Override

**void** run() {

System.***out***.println("Akarsh's own way of running...");

}

**void** swim() {

System.***out***.println("Akarsh can swim really well...");

}

}

Output:

Generic way of eating...

Generic way of how Humans breathe...

Aishwarya's own way of running...

-------------------------------

Akarsh's own way of eating...

Generic way of how Humans breathe...

Akarsh's own way of running...

Akarsh can swim really well...

//Encupsulaton

public class Encapsulate

{

private String Name;

private int Roll;

private int Age;

public int getAge()

{

return Age;

}

public String getName()

{

return Name;

}

public int getRoll()

{

return Roll;

}

public void setAge( int newAge)

{

Age = newAge;

}

public void setName(String newName)

{

Name = newName;

}

public void setRoll( int newRoll)

{

Roll = newRoll;

}

}

public class TestEncapsulation

{

public static void main (String[] args)

{

Encapsulate obj = new Encapsulate();

obj.setName("Harsh");

obj.setAge(19);

obj.setRoll(51);

System.out.println("My name: " + obj.getName());

System.out.println("My age: " + obj.getAge());

System.out.println("My roll: " + obj.getRoll());

}

}

Output:

My name:hemas

My age:22

My roll:31

//Abstruction

package com.mphasis.training.abs;

public class UnderstandingAbstractClasses {

public static void main(String[] args) {

// new Human();//We can't do this now...

Lakshmi lakshmi = new Lakshmi();

lakshmi.eat();

lakshmi.walk();

lakshmi.breathe();

new Imran().eat();

}

}

abstract class Human{

abstract void eat();

abstract void walk();

void breathe() {

System.out.println("Generic way of breathing...");

}

}

class Imran extends Human{

@Override

void eat() {

System.out.println("Imran's own way of eating...");

}

@Override

void walk() {

System.out.println("Imran's walk...");

}

}

class Lakshmi extends Human{

@Override

void eat() {

System.out.println("Lakshmi's eat...");

}

@Override

void walk() {

System.out.println("Lakhsmi's walk...");

}

}

Output:

Lakshmi's eat...

Lakhsmi's walk...

Generic way of breathing...

Imran's own way of eating...