Solution 34:

package com.hsbc.pack;

class Vehicle { // parent class vehicle

public Vehicle() {

super();

System.out.println("This is a vehicle");

}

}

class Car extends Vehicle { //car class

public Car() {

super();

}

public void printDesc() {

System.out.println("This is a car");

}

}

class Truck extends Vehicle { //truck class

public Truck() {

super();

}

public void printDesc() {

System.out.println("This is a truck");

}

}

class Bus extends Vehicle { //bus class

public Bus() {

super();

}

public void printDesc() {

System.out.println("This is a bus");

}

}

public class Road {

public static void main(String[] args) {

Car car = new Car();

car.printDesc();

Truck truck = new Truck();

truck.printDesc();

Bus bus = new Bus();

bus.printDesc();

System.out.println();

}

}

Solution 35:

package com.hsbc.pack;

class Vehicle { // parent class vehicle

String color;

int noOfWheels;

String modelName;

public Vehicle() {

super();

}

public Vehicle(String color, int noOfWheels, String modelName) { // parameterized constructor

super();

this.color = color;

this.noOfWheels = noOfWheels;

this.modelName = modelName;

}

}

// creation of base classes

class Bike extends Vehicle {

public Bike() {

super("Red",2,"Hero Honda");

}

public void details() {

System.out.println("Bike Name : " + this.modelName + "\n" + "Bike Color : " + this.color + "\n" + "No. of Wheels : " + this.noOfWheels);

}

}

class Car extends Vehicle {

public Car() {

super("Black",4,"Lamborghini");

}

public void details() {

System.out.println("Car Name : " + this.modelName + "\n" + "Car Color : " + this.color + "\n" + "No. of Wheels : " + this.noOfWheels);

}

}

class Truck extends Vehicle {

public Truck() {

super("White",8,"DTDC Bus");

}

public void details() {

System.out.println("Truck Name : " + this.modelName + "\n" + "Truck Color : " + this.color + "\n" + "No. of Wheels : " + this.noOfWheels);

}

}

public class SuperKeywordExample {

public static void main(String[] args) {

Bike b = new Bike();

b.details();

Car c = new Car();

c.details();

Truck t = new Truck();

t.details();

}

}

Solution 36:

package com.hsbc.pack;

class Vehicle { //parent class

private String color;

private int noOfWheels;

private String modelName;

public Vehicle() {

super();

}

public Vehicle(String color, int noOfWheels, String modelName) { // parameterized constructor

super();

this.color = color;

this.noOfWheels = noOfWheels;

this.modelName = modelName;

}

protected void display() {

System.out.println("Vehicle Type : " + this.getClass().getName() + "\n" +"Vehicle Model" + this.modelName + "\n" + "Vehicle color : " + this.color + "\n" + "Vehicle no. of wheels : " + this.noOfWheels);

}

}

class Bike extends Vehicle {

public Bike() {

super("Red",2,"Hayabusa");

}

}

class Car extends Vehicle {

public Car() {

super("White",4,"Lamborghini");

}

}

class Truck extends Vehicle {

public Truck() {

super("Black",8,"DTDC truck");

}

}

class City extends Vehicle {

public static void main(String[] args) {

Bike b = new Bike();

b.display();

Car c = new Car();

c.display();

Truck t = new Truck();

t.display();

}

}

Solution 37:

package com.hsbc.pack;

public class Tiger {

String color;

int weight, age;

public Tiger(String color, int weight, int age) {

super();

System.out.println("This is a tiger");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Tiger is not vegetarian");

}

public void canClimb() {

System.out.println("Tiger can climb trees");

}

public void sound() {

System.out.println("Tigers growls");

}

}

final class Lion {

String color;

int weight,age;

public Lion(String color, int weight, int age) {

super();

System.out.println("This is a lion");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Lion is not vegetarian");

}

public void canClimb() {

System.out.println("Lions cannot climb trees");

}

public void sound() {

System.out.println("Lion roars");

}

}

final class Deer {

String color;

int weight, age;

public Deer(String color, int weight, int age) {

super();

System.out.println("This is a deer");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Deers are vegetarians");

}

public void canClimb() {

System.out.println("Deers cannot climb trees");

}

public void sound() {

System.out.println("Deer grunts");

}

}

public class Animal {

public void isVegetarian() {

System.out.println("Animals exist in different classes of diet: Herbivores | Carnivores | Omnivores ");

}

public void canClimb() {

System.out.println("A few animals are climb trees");

}

public Animal() {

super();

}

public static void main(String[] args) {

Lion lion =new Lion("ABC",140,19);

System.out.print("Color Information : ");

lion.getColor();

System.out.print("Age Information : ");

lion.getAge();

System.out.print("Weight Information : ");

lion.getWeight();

System.out.print("Capability to climb : ");

lion.canClimb();

System.out.print("Vegetarian or not : ");

lion.isVegetarian();

System.out.print("Animal sound : ");

lion.sound();

Tiger tiger =new Tiger("Orange",140,19);

System.out.print("Color Information : ");

tiger.getColor();

System.out.print("Age Information : ");

tiger.getAge();

System.out.print("Weight Information : ");

tiger.getWeight();

System.out.print("Capability to climb : ");

tiger.canClimb();

System.out.print("Vegetarian or not : ");

tiger.isVegetarian();

System.out.print("Animal sound : ");

tiger.sound();

Deer deer =new Deer("Ochre",50,22);

System.out.print("Color Information : ");

deer.getColor();

System.out.print("Age Information : ");

deer.getAge();

System.out.print("Weight Information : ");

deer.getWeight();

System.out.print("Capability to climb : ");

deer.canClimb();

System.out.print("Vegetarian or not : ");

deer.isVegetarian();

System.out.print("Animal sound : ");

deer.sound();

}

Solution 38:

package com.hsbc.pack;

final class Tiger {

String color;

int weight, age;

public Tiger(String color, int weight, int age) {

super();

System.out.println("This is a tiger");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Tiger is not vegetarian");

}

public void canClimb() {

System.out.println("Tiger can climb trees");

}

public void sound() {

System.out.println("Tigers growls");

}

}

final class Lion {

String color;

int weight,age;

public Lion(String color, int weight, int age) {

super();

System.out.println("This is a lion");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Lion is not vegetarian");

}

public void canClimb() {

System.out.println("Lions cannot climb trees");

}

public void sound() {

System.out.println("Lion roars");

}

}

final class Deer {

String color;

int weight, age;

public Deer(String color, int weight, int age) {

super();

System.out.println("This is a deer");

this.color = color;

this.weight = weight;

this.age = age;

}

public void getColor() {

System.out.println(this.color);

}

public void getWeight() {

System.out.println(this.weight + " kgs.");

}

public void getAge() {

System.out.println(this.age);

}

public void isVegetarian() {

System.out.println("Deers are vegetarians");

}

public void canClimb() {

System.out.println("Deers cannot climb trees");

}

public void sound() {

System.out.println("Deer grunts");

}

}

public class Animal {

public void isVegetarian() {

System.out.println("Animals exist in different classes of diet: Herbivores | Carnivores | Omnivores ");

}

public void canClimb() {

System.out.println("A few animals are climb trees");

}

public Animal() {

super();

}

public static void main(String[] args) {

Lion lion =new Lion("ABC",140,19);

System.out.print("Color Information : ");

lion.getColor();

System.out.print("Age Information : ");

lion.getAge();

System.out.print("Weight Information : ");

lion.getWeight();

System.out.print("Capability to climb : ");

lion.canClimb();

System.out.print("Vegetarian or not : ");

lion.isVegetarian();

System.out.print("Animal sound : ");

lion.sound();

Tiger tiger =new Tiger("Orange",140,19);

System.out.print("Color Information : ");

tiger.getColor();

System.out.print("Age Information : ");

tiger.getAge();

System.out.print("Weight Information : ");

tiger.getWeight();

System.out.print("Capability to climb : ");

tiger.canClimb();

System.out.print("Vegetarian or not : ");

tiger.isVegetarian();

System.out.print("Animal sound : ");

tiger.sound();

Deer deer =new Deer("Ochre",50,22);

System.out.print("Color Information : ");

deer.getColor();

System.out.print("Age Information : ");

deer.getAge();

System.out.print("Weight Information : ");

deer.getWeight();

System.out.print("Capability to climb : ");

deer.canClimb();

System.out.print("Vegetarian or not : ");

deer.isVegetarian();

System.out.print("Animal sound : ");

deer.sound();

}

Solution 39:

package com.hsbc.pack;

import java.util.Scanner;

public class Worker { //parent class

String name;

double salaryRate;

public String getName() { /\* getters and setters

return name;

}

public void setName(String name) {

this.name = name;

}

public double getSalaryRate() {

return salaryRate;

}

public void setSalaryRate(double salaryRate) { \*/

this.salaryRate = salaryRate;

}

public static void main(String[] args) {

DailyWorker dailyWorker = new DailyWorker();

dailyWorker.pay();

SalariedWorker salariedWorker = new SalariedWorker();

salariedWorker.pay();

}

}

class DailyWorker extends Worker { //child class

Scanner sc = new Scanner(System.in);

void pay () {

System.out.println("Daily Worker Salary");

DailyWorker dailyWorker = new DailyWorker();

System.out.println("Enter total working days : ");

int hrs = sc.nextInt();

System.out.println("Enter Daily Wage : ");

dailyWorker.setSalaryRate(sc.nextDouble());

System.out.println("Total Salary is " + (hrs\*dailyWorker.getSalaryRate()));

}

}

class SalariedWorker extends Worker{ //child class

Scanner sc = new Scanner(System.in);

void pay () {

System.out.println("Salaried Worker Salary");

SalariedWorker salariedWorker = new SalariedWorker();

System.out.println("Enter your per hour salary: ");

salariedWorker.setSalaryRate(sc.nextDouble());

System.out.println("Total Salary is " + (40\*salariedWorker.getSalaryRate()));

}

}

Solution 40:

package com.hsbc.pack;

Solution 41:

package com.hsbc.pack;

import java.io.\*;

abstract class Shape{ //abstract class

abstract public void surfaceArea();

abstract public void Volume();

final float pi=3.14f;

}

class Sphere extends Shape{

double r;

private double area;

private double volume;

public void accept() throws IOExecption{ // accept() to take input

System.out.println("Enter the radius of the Sphere: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=pi\*r\*r;

}

public void Volume(){

volume=1.3334\*pi\*r\*r\*r; // 4/3=1.3334

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of sphere is: "+area);

System.out.println("The volume of sphere is: "+volume);

}

}

class Cone extends Shape{

double h,r,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter radius and height of the Cone: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

double sq=h\*h+r\*r;

area=pi\*r\*(r+java.lang.Math.sqrt(sq));

}

public void Volume(){

double d=h/3;

volume=pi\*r\*r\*d;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Cone is: "+area);

System.out.println("The volume of Cone is: "+volume);

}

}

class Cylinder extends Shape{

double r,h,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter radius and height of the Cylinder: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=(2\*pi\*r\*h)+(2\*pi\*r\*r);

}

public void Volume(){

volume=pi\*r\*r\*h;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Cylinder is: "+area);

System.out.println("The volume of Cylinder is: "+volume);

}

}

class Box extends Shape{

double l,b,h,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter length, breadth and height of the Box: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

l=Double.parseDouble(br.readLine());

b=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=(2\*l\*b)+(2\*b\*h)+(2\*l\*h);

}

public void Volume(){

volume=l\*b\*h;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Box is: "+area);

System.out.println("The volume of Box is: "+volume);

}

}

public class ShapesDemo {

public static void main(String [] args) throws IOException{

Sphere s=new Sphere();

s.accept();

s.display();

Cone co=new Cone();

co.accept();

co.display();

Cylinder cy=new Cylinder();

cy.accept();

cy.display();

Box b=new Box();

b.accept();

b.display();

}

}

Solution 42:

import java.util.Arrays;

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class Course extends Menu {

static Set<Integer> set = new HashSet<>(Arrays.asList());

static Course[] course = {

new Course(1, "JS", 3000, "Online", "Full-time" ),

new Course(2, "Python", 4000, "Classroom", "Part-time" ) ,

new Course(3, "C++", 5000, "Online", "Part-time" ),

new Course(4, "Java", 6000, "Classroom", "Full-time")

};

Employee[] employees = {

new Employee (1, "Aman"),

new Employee (2, "navya"),

new Employee (3, "ruhika"),

new Employee (4, "utkarsh"),

new Employee (5, "gaurav")

};

static Course Solution = new Course();

public static void main(String[] args) {

Scanner scnr = new Scanner(System.in);

System.out.println("1.To register for a course\r\n" +

"2.To deregister for a course\r\n" +

"3.To list the courses offered\r\n" +

"4.To list the employees registered for a specific course.");

System.out.println("Enter option number you want to go for : ");

int i = scnr.nextInt();

int eID = 0;

int cID = 0;

switch (i) {

case 1 :

System.out.println("Enter Course ID and Employee ID");

cID = scnr.nextInt();

eID = scnr.nextInt();

Solution.registerForCourse(cID-1, eID);

break;

case 2:

System.out.println("Enter Course ID");

cID = scnr.nextInt();

System.out.println(course[cID-1]);

System.out.println("Enter Employee ID");

eID = scnr.nextInt();

Solution.deregisterForCourse(cID-1, eID);

break;

case 3:

Solution.listOfferedCourse();

break;

case 4:

System.out.println("Enter Course number to find all Employees who is enrolled :");

Solution.listEmployeesOfCourse((scnr.nextInt()) - 1);

break;

}

main(args);

scnr.close();

}

@Override

void registerForCourse(int courseID, int employeeID) {

// TODO Auto-generated method stub

course[courseID].employeeIDs.add(employeeID);

System.out.println("Updated Course Structure");

System.out.println(course[courseID]);

}

@Override

void deregisterForCourse(int courseID, int employeeID) {

// TODO Auto-generated method stub

course[courseID].employeeIDs.remove(employeeID);

System.out.println("Updated Course Structure");

System.out.println(course[courseID]);

}

@Override

String[] listOfferedCourse() {

for(Course curs : course) {

System.out.println(curs);

}

return null;

}

@Override

String[] listEmployeesOfCourse(int courseID) {

Set<Integer> emps = course[courseID].employeeIDs;

if(emps.size() != 0)

{

for(int empID : emps ) {

System.out.println(employees[empID-1].toString());

}

}

else {

System.out.println("No Employee is registered for this course");

}

return null;

}

}

abstract class Menu {

abstract void registerForCourse(int courseID, int employeeID);

abstract void deregisterForCourse(int courseID, int employeeID);

abstract String[] listOfferedCourse();

abstract String[] listEmployeesOfCourse(int courseID);

}

class Course{

int courseID;

String courseName;

double courseFee;

String courseDeliveryType;

String courseDurationType;

Set<Integer> employeeIDs = new HashSet<Integer>();

public Course(int courseID, String courseName, double courseFee, String courseDeliveryType,

String courseDurationType) {

super();

this.courseID = courseID;

this.courseName = courseName;

this.courseFee = courseFee;

this.courseDeliveryType = courseDeliveryType;

this.courseDurationType = courseDurationType;

this.employeeIDs = employeeIDs;

}

@Override

public String toString() {

return "Course [courseID=" + courseID + ", courseName=" + courseName + ", courseFee=" +courseFee+ ", courseDeliveryType=" + courseDeliveryType + ", courseDurationType=" + courseDurationType+ ", employeeIDs=" + employeeIDs + "]";

}

}

class Employee{

int empID;

String empName;

public Employee(int empID, String empName) {

super();

this.empID = empID;

this.empName = empName;

}

public Employee() {

super();

}

@Override

public String toString() {

return "Employee [empID=" + empID + ", empName=" + empName + "]";

}

}

Solution 43:

package com.hsbc.pack;

import java.io.\*;

interface Shape{ //abstract class

abstract public void surfaceArea();

abstract public void Volume();

float pi=3.14f;

}

class Sphere implements Shape{

double r;

private double area;

private double volume;

public void accept() throws IOExecption{ // accept() to take input

System.out.println("Enter the radius of the Sphere: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=pi\*r\*r;

}

public void Volume(){

volume=1.3334\*pi\*r\*r\*r; // 4/3=1.3334

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of sphere is: "+area);

System.out.println("The volume of sphere is: "+volume);

}

}

class Cone implements Shape{

double h,r,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter radius and height of the Cone: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

double sq=h\*h+r\*r;

area=pi\*r\*(r+java.lang.Math.sqrt(sq));

}

public void Volume(){

double d=h/3;

volume=pi\*r\*r\*d;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Cone is: "+area);

System.out.println("The volume of Cone is: "+volume);

}

}

class Cylinder implements Shape{

double r,h,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter radius and height of the Cylinder: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

r=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=(2\*pi\*r\*h)+(2\*pi\*r\*r);

}

public void Volume(){

volume=pi\*r\*r\*h;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Cylinder is: "+area);

System.out.println("The volume of Cylinder is: "+volume);

}

}

class Box implements Shape{

double l,b,h,area,volume;

public void accept() throws IOExecption{

System.out.println("Enter length, breadth and height of the Box: ");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

l=Double.parseDouble(br.readLine());

b=Double.parseDouble(br.readLine());

h=Double.parseDouble(br.readLine());

}

public void surfaceArea(){

area=(2\*l\*b)+(2\*b\*h)+(2\*l\*h);

}

public void Volume(){

volume=l\*b\*h;

}

public void display(){

surfaceArea();

Volume();

System.out.println("The area of Box is: "+area);

System.out.println("The volume of Box is: "+volume);

}

}

public class ShapesDemo {

public static void main(String [] args) throws IOException{

Sphere s=new Sphere();

s.accept();

s.display();

Cone co=new Cone();

co.accept();

co.display();

Cylinder cy=new Cylinder();

cy.accept();

cy.display();

Box b=new Box();

b.accept();

b.display();

}

}

Solution 44:

package com.hsbc.pack;

public class Drawing {

public static void main(String[] args) {

Line line = new Line();

line.drawingColor();

line.fillingColor();

line.size();

line.thickness();

System.out.println();

Circle circle = new Circle();

circle.drawingColor();

circle.fillingColor();

circle.size();

circle.thickness();

System.out.println();

Square square = new Square();

square.drawingColor();

square.fillingColor();

square.size();

square.thickness();

}

}

interface IDrawable { // first interface

void drawingColor();

void thickness();

}

interface IFillable{ // second interface

void fillingColor();

void size();

}

class Line implements IDrawable, IFillable {

@Override

public void fillingColor() {

// TODO Auto-generated method stub

System.out.println("fillingColor in Line");

}

@Override

public void size() {

// TODO Auto-generated method stub

System.out.println("size in Line");

}

@Override

public void drawingColor() {

// TODO Auto-generated method stub

System.out.println("drawingColor in Line");

}

@Override

public void thickness() {

// TODO Auto-generated method stub

System.out.println("thickness in Line");

}

}

class Circle implements IDrawable, IFillable {

@Override

public void fillingColor() {

// TODO Auto-generated method stub

System.out.println("fillingColor in Circle");

}

@Override

public void size() {

// TODO Auto-generated method stub

System.out.println("size in Circle");

}

@Override

public void drawingColor() {

// TODO Auto-generated method stub

System.out.println("drawingColor in Circle");

}

@Override

public void thickness() {

// TODO Auto-generated method stub

System.out.println("thickness in Circle");

}

}

class Square implements IDrawable, IFillable {

@Override

public void fillingColor() {

// TODO Auto-generated method stub

System.out.println("fillingColor in Square");

}

@Override

public void size() {

// TODO Auto-generated method stub

System.out.println("size in Square");

}

@Override

public void drawingColor() {

// TODO Auto-generated method stub

System.out.println("drawingColor in Square");

}

@Override

public void thickness() {

// TODO Auto-generated method stub

System.out.println("thickness in Square");

}

}

Solution 45:

package com.hsbc.pack;

import java.util.Scanner;

interface Payable {

void getPayInfo();

}

public class Wages

{

public static void main(String[] args) {

DailyWorker dailyWorker = new DailyWorker();

dailyWorker.pay();

SalariedWorker salariedWorker = new SalariedWorker();

salariedWorker.pay();

}

}

class DailyWorker implements Payable { //child class

Scanner sc = new Scanner(System.in);

void pay () {

System.out.println("Daily Worker Salary");

DailyWorker dailyWorker = new DailyWorker();

System.out.println("Enter total working days : ");

int hrs = sc.nextInt();

System.out.println("Enter Daily Wage : ");

dailyWorker.setSalaryRate(sc.nextDouble());

System.out.println("Total Salary is " + (hrs\*dailyWorker.getSalaryRate()));

}

}

class SalariedWorker implements Payable{ //child class

Scanner sc = new Scanner(System.in);

void pay () {

System.out.println("Salaried Worker Salary");

SalariedWorker salariedWorker = new SalariedWorker();

System.out.println("Enter your per hour salary: ");

salariedWorker.setSalaryRate(sc.nextDouble());

System.out.println("Total Salary is " + (40\*salariedWorker.getSalaryRate()));

}

}

Solution 46:

package com.hsbc.pack;

Solution 47:

package com.hsbc.pack;

public class BankAccount {

private float balance;

// Inner class

public class InterestAdder

{

public float interestAdder()

{

return (balance+(balance\*12/100)); //adding interest to the balance amount

}

}

public float getBalance() {

InterestAdder in = new InterestAdder();

return in.interestAdder();

}

public void setBalance(float balance) {

this.balance = balance;

}

public void deposit(float amount){

this.balance += amount;

}

public void withdraw(float amount){

this.balance -= amount;

}

public static void main(String[] args) {

BankAccount obj = new BankAccount();

obj.deposit(500);

obj.withdraw(200);

obj.deposit(700);

System.out.println(obj.getBalance());

}

}

Solution 48:

package com.hsbc.pack;

public class BankAccount {

float balance;

public void setBalance(float balance){

this.balance = balance;

}

public float getBalance()

{

class Inner

{

public float interestAdder()

{

return (balance+(balance\*12/100)); //interest calculation

}

}

Inner in = new Inner();

this.balance = in.interestAdder();

return balance;

}

public void deposit(float amount){

this.balance += amount;

}

public void withdraw(float amount){

this.balance -= amount;

}

public static void main(String[] args) {

BankAccount obj = new BankAccount();

obj.deposit(50000);

System.out.println(obj.getBalance()); // getting balance

}

}

Solution 49:

package com.hsbc.pack;

interface Calculate{

int interest();

}

class BankAccount implements Calculate{

public int balance, interest;

void deposit(int x) {

balance = balance+x; // adding in balance

}

void withdraw(int x) {

//withdrawing from the balance

balance = balance-x;

}

int getBalance() { // to get sum of deposit and interest over it

Calculate p=new Calculate(){

int interest()

{

return(((balance)\*12)/(100));

}

};

balance=balance+p.interest();

return balance;

}

public static void main(String[] args)

{

BankAccount b=new BankAccount();

b.deposit(200);

System.out.println(b.getBalance());

}

}