**Exercise 34:** *Create a class called Vehicle. Create subclasses like Truck, Bus, Car etc. Add common methods*

*in the base class and specific methods in the corresponding class. Create a class called Road*

*and create objects for the Truck, Car, Bus etc and display the appropriate message.*

**Solution 34**

**package** com.hsbc.day3;

/\*Create a class called Vehicle. Create subclasses like Truck, Bus, Car etc. Add common methods

in the base class and specific methods in the corresponding class. Create a class called Road

and create objects for the Truck, Car, Bus etc and display the appropriate message.\*/

**public** **class** Road {

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

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

Truck t=**new** Truck();

t.color();

t.name();

Bus b=**new** Bus();

b.color();

b.name();

Car c=**new** Car();

c.color();

c.name();

}

}

**class** Vehicle{

**public** **void** color() {

}

**public** **void** name() {

}

}

//Extending the vehicle class

**class** Truck **extends** Vehicle{

@Override

**public** **void** color() {

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

**super**.color();

System.***out***.println("The color of Truck is blue.");

}

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle name: Truck");

}

}

//Extending the vehicle class

**class** Bus **extends** Vehicle{

@Override

**public** **void** color() {

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

**super**.color();

System.***out***.println("The color of the Bus is blue.");

}

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle Name: Bus");

}

}

//Extending the vehicle class

**class** Car **extends** Vehicle{

@Override

**public** **void** color() {

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

**super**.color();

System.***out***.println("The color of the car is Red.");

}

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle Name: Car");

}

}

**Output:**

The color of Truck is blue.

Vehicle name: Truck

The color of the Bus is blue.

Vehicle Name: Bus

The color of the car is Red.

Vehicle Name: Car

**Exercise 35:** *In the Lab Exercise 34, in the Vehicle class constructor initialize few variables like color, no of*

*wheels, model etc. Give appropriate values for these variables from the invoking subclass.*

**Solution 35:**

**package** com.hsbc.day3;

/\*Invoking the sub classes by providing the value for parameterized constructor of super class.\*/

**public** **class** Road {

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

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

Truck t=**new** Truck();

t.name();

t.desc();

Bus b=**new** Bus();

b.name();

b.desc();

Car c=**new** Car();

c.name();

c.desc();

}

}

//Declaration of a class

**class** Vehicle{

**private** String color;

**private** **int** noOfWheels;

**private** String model;

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

**public** **int** getNoOfWheels() {

**return** noOfWheels;

}

**public** **void** setNoOfWheels(**int** noOfWheels) {

**this**.noOfWheels = noOfWheels;

}

**public** String getModel() {

**return** model;

}

**public** **void** setModel(String model) {

**this**.model = model;

}

**public** **void** name() {

}

//Parameterised constructor

**public** Vehicle(String color, **int** noOfWheels, String model) {

**super**();

**this**.color = color;

**this**.noOfWheels = noOfWheels;

**this**.model = model;

}

//Default constructor

**public** Vehicle() {

**super**();

}

//Function to print the details about vehicle

**public** **void** desc() {

System.***out***.println("Color: "+**this**.color+" No. of wheels: "+**this**.noOfWheels+" Model: "+**this**.model);

}

}

//Extending the vehicle class

**class** Truck **extends** Vehicle{

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle name: Truck");

}

**public** Truck() {

//calling the constructor of super class

**super**("Blue",6,"V5");

}

}

//Extending the vehicle class

**class** Bus **extends** Vehicle{

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle Name: Bus");

}

**public** Bus() {

//calling the constructor of super class

**super**("Red",6,"b5");

}

}

//Extending the vehicle class

**class** Car **extends** Vehicle{

@Override

**public** **void** name() {

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

**super**.name();

System.***out***.println("Vehicle Name: Car");

}

//calling the constructor of super class

**public** Car() {

**super**("White",4,"Swift");

}

}

**Output:**

Vehicle name: Truck

Color: Blue No. of wheels: 6 Model: V5

Vehicle Name: Bus

Color: Red No. of wheels: 6 Model: b5

Vehicle Name: Car

Color: White No. of wheels: 4 Model: Swift

**Exercise 36:** *In the Lab Exercise 35, create another class called City which creates an object for the Car,*

*Truck and Bus class and displays the details through a display () method in the Vehicle class.*

*The other methods and data members should not be accessible by the City class.*

**Exercise 39:** *Create a class called Worker. Write classes DailyWorker and SalariedWorker that inherit from Worker. Every worker has a name and a salaryrate. Write method Pay (int hours) to compute*

*the week pay of every worker. A Daily worker is paid on the basis of the number of days*

*she/he works.The salaried worker gets paid the wage for 40 hours a week no matter what the*

*actual hours are. Test this program to calculate the pay of workers.*

**Solution:**

**package** com.hsbc.day3;

**import** java.util.Scanner;

//Create a class and perform polymorphism functions.

**public** **class** Solution39 {

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

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

Scanner sc=**new** Scanner(System.***in***);

//Creating a object

DailyWorker d=**new** DailyWorker();

System.***out***.println("Enter your name");

d.setName(sc.next());

System.***out***.println("Enter your Salary Rate");

d.setSalaryRate(sc.nextDouble());

System.***out***.println("Enter number of hours: ");

**int** hours=sc.nextInt();

System.***out***.println("Name: "+d.getName());

d.pay(hours);

SalariedWorker s=**new** SalariedWorker();

System.***out***.println("Enter your name");

s.setName(sc.next());

System.***out***.println("Enter your Salary Rate");

s.setSalaryRate(sc.nextDouble());

System.***out***.println("Name: "+s.getName());

s.pay(40);

sc.close();

}

}

**class** Worker{

**private** String name;

**private** **double** salaryRate;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalaryRate() {

**return** salaryRate;

}

**public** **void** setSalaryRate(**double** salaryRate) {

**this**.salaryRate = salaryRate;

}

//declaring function

**public** **void** pay(**int** hours)

{

}

}

**class** DailyWorker **extends** Worker{

//overriding the function

**public** **void** pay(**int** hours)

{

**double** total=getSalaryRate()\*hours;

System.***out***.println("The salary is: "+total);

}

}

**class** SalariedWorker **extends** Worker{

**public** **void** pay(**int** hours)

{

**double** total=getSalaryRate()\*hours;

System.***out***.println("The salary is: "+total);

}

}

**Output:**

Enter your name

John

Enter your Salary Rate

22

Enter number of hours:

43

Name: John

The salary is: 946.0

Enter your name

Ram

Enter your Salary Rate

26

Name: Ram

The salary is: 1040.0

**Exercise 41:** *Create a class called Shape3D with the following method signatures alone, volume () and*

*surfaceArea (). Then create subclasses like Cylinder, Sphere, and Cubeetc and implement*

*these methods.*

**Solution 41:**

**package** com.hsbc.day3;

**import** java.lang.Math;

/\*Overriding the methods of abstract class

and calculating the volume and surface area.\*/

**public** **class** Solution41 {

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

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

Cylinder c=**new** Cylinder(2.3,1.5);

c.surfaceArea();

c.volume();

Cube c1=**new** Cube(5);

c1.surfaceArea();

c1.volume();

}

}

//Abstract class creation

**abstract** **class** Shape3D{

//Abstract method creation

**public** **abstract** **void** volume();

**public** **abstract** **void** surfaceArea();

}

//Inheriting the abstract class

**class** Cylinder **extends** Shape3D{

**private** **double** radius;

**private** **double** height;

**public** **double** getRadius() {

**return** radius;

}

**public** **void** setRadius(**double** radius) {

**this**.radius = radius;

}

**public** **double** getHeight() {

**return** height;

}

**public** **void** setHeight(**double** height) {

**this**.height = height;

}

**public** Cylinder(**double** radius,**double** height) {

**this**.radius=radius;

**this**.height=height;

}

//Implementing the abstract method volume

**public** **void** volume()

{

**double** volume=Math.***PI***\*radius\*radius\*height;

System.***out***.printf("The volume of cylinder is: %.2f\n", volume);

}

//Implementing the abstract method surface area

**public** **void** surfaceArea()

{

**double** area=Math.***PI***\*2\*radius\*height;

System.***out***.printf("The Surface area of cylinder is: %.2f\n", area);

}

}

//Inheriting the abstract class

**class** Cube **extends** Shape3D{

**private** **int** side;

**public** **int** getSide() {

**return** side;

}

**public** **void** setSide(**int** side) {

**this**.side = side;

}

**public** Cube(**int** side) {

**this**.side=side;

}

//Implementing the abstract method volume

**public** **void** volume()

{

**double** volume=side\*side\*side;

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

}

//Implementing the abstract method surface area

**public** **void** surfaceArea()

{

**double** area=4\*side;

System.***out***.println("The Surface area of cube is: "+area);

}

}

**Output:**

The Surface area of cylinder is: 21.68

The volume of cylinder is: 24.93

The Surface area of cube is: 20.0

The volume of cube is: 125.0

**Exercise 42:** *Create the classes required to store data regarding different types of courses that employees Ina company can enroll for. All courses have name and course fee. Courses are also either*

*classroom delivered or delivered online. Courses could also be full time or part time. The*

*program must be menu based input which enables the course coordinator to register*

*employees for courses, list out employees registered for specific courses, deregister employees*

*from a course.*

**Solution:**

**Class Course:**

/\*\*

\*

\*/

package com.hsbc.course;

import java.util.ArrayList;

/\*\*

\* @author Sunrise

\* Declaring the abstract class with all the methods.

\*

\*/

public abstract class Course {

abstract void registerCourse(Employee e);

abstract void deregisterCourse(Employee e);

abstract ArrayList<CourseDetails> listCourses();

abstract ArrayList<Employee> listEmployee();

abstract ArrayList<CourseDetails> addCourse(CourseDetails c);

}

**Class CourseDetails:**

**package** com.hsbc.course;

/\*Decalring the course details variables which are needed.\*/

**public** **class** CourseDetails {

**private** String courseName;

**private** String mode;

**private** String time;

**private** **int** fee;

**public** String getCourseName() {

**return** courseName;

}

**public** **void** setCourseName(String courseName) {

**this**.courseName = courseName;

}

**public** String getMode() {

**return** mode;

}

**public** **void** setMode(String mode) {

**this**.mode = mode;

}

**public** String getTime() {

**return** time;

}

**public** **void** setTime(String time) {

**this**.time = time;

}

**public** **int** getFee() {

**return** fee;

}

**public** **void** setFee(**int** fee) {

**this**.fee = fee;

}

**public** CourseDetails(String courseName, String mode, String time, **int** fee) {

**super**();

**this**.courseName = courseName;

**this**.mode = mode;

**this**.time = time;

**this**.fee = fee;

}

**public** CourseDetails() {

**super**();

}

@Override

**public** String toString() {

**return** "CourseDetails [courseName=" + courseName + ", mode=" + mode + ", time=" + time + ", fee=" + fee + "]";

}

}

**Class Employee:**

**package** com.hsbc.course;

//Declaring the variables related to mployee and extending the CourseDetails class.

**public** **class** Employee **extends** CourseDetails{

**private** String ename;

**private** **int** eid;

**public** String getEname() {

**return** ename;

}

**public** **void** setEname(String ename) {

**this**.ename = ename;

}

**public** **int** getEid() {

**return** eid;

}

**public** **void** setEid(**int** eid) {

**this**.eid = eid;

}

**public** Employee() {

**super**();

}

//Parameterized constructor with all the variables of both the classes.

**public** Employee(String courseName, String mode, String time, **int** fee, String ename, **int** eid) {

**super**(courseName, mode, time, fee);

**this**.ename = ename;

**this**.eid = eid;

}

@Override

**public** String toString() {

**return** "Employee [ename=" + ename + ", eid=" + eid + "]";

}

}

**Class Impl:**

**package** com.hsbc.course;

**import** java.util.ArrayList;

//Extending the abstract class and implementing all the methods.

**public** **class** Impl **extends** Course {

//Declaring array lists for storage.

ArrayList<Employee> details=**new** ArrayList<Employee>();

ArrayList<CourseDetails> courseDetails=**new** ArrayList<CourseDetails>();

@Override

**void** registerCourse(Employee e) {

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

details.add(e);

}

@Override

**void** deregisterCourse(Employee e) {

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

**if**(details.get(e.getEid())!=**null**)

details.remove(e);

**else**

System.***out***.println("No Matches");

}

@Override

ArrayList<CourseDetails> listCourses() {

**if**(courseDetails.isEmpty())

System.***out***.println("No courses");

**else**

System.***out***.println(courseDetails);

**return** **null**;

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

}

@Override

ArrayList<Employee> listEmployee() {

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

**if**(details.isEmpty())

System.***out***.println("No Employee has enrolled yet.");

**else**

System.***out***.println(details);

**return** **null**;

}

@Override

ArrayList<CourseDetails> addCourse(CourseDetails c) {

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

courseDetails.add(c);

**return** **null**;

}

}

**Class Manipulation:**

**package** com.hsbc.course;

**import** java.util.\*;

//calling all the functions.

**public** **class** Manipulation {

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

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

Scanner sc=**new** Scanner(System.***in***);

//adding some courses initially.

CourseDetails c=**new** CourseDetails("ML","Classroom","Part-Time",10000);

CourseDetails c1=**new** CourseDetails("AI","Classroom","Part-Time",10000);

CourseDetails c2=**new** CourseDetails("DS","Classroom","Part-Time",10000);

CourseDetails c3=**new** CourseDetails("DA","Classroom","Part-Time",10000);

Impl i11=**new** Impl();

i11.addCourse(c1);

i11.addCourse(c3);

i11.addCourse(c);

i11.addCourse(c2);

//i11.listCourses();

**while**(**true**) {

//Menu for the choices.

System.***out***.println("Choose from the Menu\n1. To register for a course\n2. To deregister for a course\n3. To list the courses offered\n4. To list the employees registered for a specific course.\n5. Exit.");

System.***out***.println("Enter your choice: ");

**int** choice=sc.nextInt();

**switch**(choice) {

**case** 1:

Employee e= **new** Employee("ML","Classroom","Part-Time",10000,"Ram",101);

i11.registerCourse(e);

**break**;

**case** 2:

Employee e1= **new** Employee("ML","Classroom","Part-Time",10000,"Ram",101);

i11.deregisterCourse(e1);

**break**;

**case** 3:

i11.listCourses();

**break**;

**case** 4:

i11.listEmployee();

**break**;

**case** 5:

System.*exit*(0);

}

}

}

}

**Output:**

Choose from the Menu

1. To register for a course

2. To deregister for a course

3. To list the courses offered

4. To list the employees registered for a specific course.

5. Exit.

Enter your choice:

1

Choose from the Menu

1. To register for a course

2. To deregister for a course

3. To list the courses offered

4. To list the employees registered for a specific course.

5. Exit.

Enter your choice:

4

[Employee [ename=Ram, eid=101]]

Choose from the Menu

1. To register for a course

2. To deregister for a course

3. To list the courses offered

4. To list the employees registered for a specific course.

5. Exit.

Enter your choice:

3

[CourseDetails [courseName=AI, mode=Classroom, time=Part-Time, fee=10000], CourseDetails [courseName=DA, mode=Classroom, time=Part-Time, fee=10000], CourseDetails [courseName=ML, mode=Classroom, time=Part-Time, fee=10000], CourseDetails [courseName=DS, mode=Classroom, time=Part-Time, fee=10000]]

Choose from the Menu

1. To register for a course

2. To deregister for a course

3. To list the courses offered

4. To list the employees registered for a specific course.

5. Exit.

Enter your choice:

**Exercise 43:** *Implement Lab Exercise 41using Interfaces.*

**Solution 43:**

**package** com.hsbc.day3;

**import** java.lang.Math;

/\*Implementing the interface

and calculating the volume and surface area.\*/

**public** **class** Solution43 {

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

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

Cylinder1 c=**new** Cylinder1(2.3,1.5);

c.surfaceArea();

c.volume();

Cube1 c1=**new** Cube1(5);

c1.surfaceArea();

c1.volume();

}

}

//Interface creation

**interface** Shape3D1{

//Abstract method creation

**public** **void** volume();

**public** **void** surfaceArea();

}

//Implementing the interface

**class** Cylinder1 **implements** Shape3D1{

**private** **double** radius;

**private** **double** height;

**public** **double** getRadius() {

**return** radius;

}

**public** **void** setRadius(**double** radius) {

**this**.radius = radius;

}

**public** **double** getHeight() {

**return** height;

}

**public** **void** setHeight(**double** height) {

**this**.height = height;

}

**public** Cylinder1(**double** radius,**double** height) {

**this**.radius=radius;

**this**.height=height;

}

//Implementing the abstract method volume

**public** **void** volume()

{

**double** volume=Math.***PI***\*radius\*radius\*height;

System.***out***.printf("The volume of cylinder is: %.2f\n", volume);

}

//Implementing the abstract method surface area

**public** **void** surfaceArea()

{

**double** area=Math.***PI***\*2\*radius\*height;

System.***out***.printf("The Surface area of cylinder is: %.2f\n", area);

}

}

//Implementing the interface

**class** Cube1 **implements** Shape3D1{

**private** **int** side;

**public** **int** getSide() {

**return** side;

}

**public** **void** setSide(**int** side) {

**this**.side = side;

}

**public** Cube1(**int** side) {

**this**.side=side;

}

//Implementing the abstract method volume

**public** **void** volume()

{

**double** volume=side\*side\*side;

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

}

//Implementing the abstract method surface area

**public** **void** surfaceArea()

{

**double** area=4\*side;

System.***out***.println("The Surface area of cube is: "+area);

}

}

**Output:**

The Surface area of cylinder is: 21.68

The volume of cylinder is: 24.93

The Surface area of cube is: 20.0

The volume of cube is: 125.0

**Exercise 44:** *Create two interfaces namely Drawable and Fillable. Create class called Line, Circle, Square and implement following methods through interface.*

**Solution 44:**

**package** com.hsbc.day3;

//Implementing more than one interface.

**public** **class** Solution44 {

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

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

Line l=**new** Line();

l.drawingColor();

l.fillingColor();

l.thickness();

l.size();

Circle c=**new** Circle();

c.drawingColor();

c.fillingColor();

c.thickness();

c.size();

Square s=**new** Square();

s.drawingColor();

s.fillingColor();

s.thickness();

s.size();

}

}

**interface** Drawable{

**public** **abstract** **void** drawingColor();

**public** **abstract** **void** thickness();

}

**interface** Fillable{

**public** **abstract** **void** fillingColor();

**public** **abstract** **void** size();

}

//Implementing more than one interfaces

**class** Line **implements** Drawable,Fillable{

**public** **void** drawingColor()

{

System.***out***.println("Drawing the Line with blue color.");

}

**public** **void** thickness()

{

System.***out***.println("The thickness of line is 3px");

}

@Override

**public** **void** fillingColor() {

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

System.***out***.println("The filling color of line is grey.");

}

@Override

**public** **void** size() {

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

System.***out***.println("The size of the line is unknown.");

}

}

//Implementing more than one interfaces

**class** Circle **implements** Drawable,Fillable{

**public** **void** drawingColor()

{

System.***out***.println("Drawing the circle with blue color.");

}

**public** **void** thickness()

{

System.***out***.println("The thickness of circle is 3px");

}

@Override

**public** **void** fillingColor() {

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

System.***out***.println("The filling color of circle is grey.");

}

@Override

**public** **void** size() {

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

System.***out***.println("The size of the circle is unknown.");

}

}

//Implementing more than one interfaces

**class** Square **implements** Drawable,Fillable{

**public** **void** drawingColor()

{

System.***out***.println("Drawing the Square with blue color.");

}

**public** **void** thickness()

{

System.***out***.println("The thickness of Square is 3px");

}

@Override

**public** **void** fillingColor() {

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

System.***out***.println("The filling color of Square is grey.");

}

@Override

**public** **void** size() {

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

System.***out***.println("The size of the Square is unknown.");

}

}

**Output:**

Drawing the Line with blue color.

The filling color of line is grey.

The thickness of line is 3px

The size of the line is unknown.

Drawing the circle with blue color.

The filling color of circle is grey.

The thickness of circle is 3px

The size of the circle is unknown.

Drawing the Square with blue color.

The filling color of Square is grey.

The thickness of Square is 3px

The size of the Square is unknown.

**Exercise 45:** *In Lab Exercise 39, create a package called finance and within it create an interface called*

*Payable.It should define the getPayInfo() methodthatall the workerclasseswillimplement.*

*Now display the details of the monthly pay of the workers.*

**Solution 45:**

**package** finance;

**import** java.util.Scanner;

/\*create a package called finance and within it create an interface called

Payable.It should define the getPayInfo()

methodthatall the workerclasseswillimplement.

Now display the details of the monthly pay of the workers.\*/

**public** **class** Solution45 {

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

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

Scanner sc=**new** Scanner(System.***in***);

//Creating a object

DailyWorker d=**new** DailyWorker();

System.***out***.println("Enter your name");

d.setName(sc.next());

System.***out***.println("Enter your Salary Rate");

d.setSalaryRate(sc.nextDouble());

System.***out***.println("Enter number of hours per day: ");

d.setHrs(sc.nextInt());;

System.***out***.println("For Daily Worker");

System.***out***.println("Name: "+d.getName());

//calling the overridden function

d.getPayInfo();

//creating a object for salaried worker

SalariedWorker s=**new** SalariedWorker();

System.***out***.println("Enter your name");

s.setName(sc.next());

System.***out***.println("Enter your Salary Rate");

s.setSalaryRate(sc.nextDouble());

System.***out***.println("Enter number of hours per day: ");

s.setHrs(sc.nextInt());

System.***out***.println("For Salaried Worker");

System.***out***.println("Name: "+s.getName());

//calling the overridden function

s.getPayInfo();

sc.close();

}

}

//Defining an interface

**interface** Payable{

**public** **abstract** **void** getPayInfo();

}

**class** Worker{

**private** String name;

**private** **double** salaryRate;

**private** **int** hrs;

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **double** getSalaryRate() {

**return** salaryRate;

}

**public** **void** setSalaryRate(**double** salaryRate) {

**this**.salaryRate = salaryRate;

}

**public** **int** getHrs() {

**return** hrs;

}

**public** **void** setHrs(**int** hrs) {

**this**.hrs = hrs;

}

}

**class** DailyWorker **extends** Worker **implements** Payable {

//overriding the function

**public** **void** getPayInfo()

{

**double** total=getSalaryRate()\*getHrs()\*30;

System.***out***.println("The salary is: "+total);

}

}

**class** SalariedWorker **extends** Worker **implements** Payable{

**public** **void** getPayInfo()

{

**double** total=getSalaryRate()\*getHrs()\*30;

System.***out***.println("The salary is: "+total);

}

}

Output:

Enter your name

Shyam

Enter your Salary Rate

22

Enter number of hours per day:

5

For Daily Worker

Name: Shyam

The salary is: 3300.0

Enter your name

Suman

Enter your Salary Rate

56

Enter number of hours per day:

6

For Salaried Worker

Name: Suman

The salary is: 10080.0

**Exercise47:** *Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create an inner class called InterestAdder and implement the interest calculations and add*

*the interest to the current balance.*

**Solution 47:**

**package** com.hsbc.day3;

/\*Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.

Create an inner class called InterestAdder and implement the interest calculations and add

the interest to the current balance.\*/

//outer class

**public** **class** BankAccount{

**private** **double** balance;

**public** **double** getBalance() {

**return** balance;

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

//Function to deposit amount

**public** **void** deposit(**int** x) {

balance=balance+x;

System.***out***.println("The changed amount after deposit is: "+ getBalance());

}

//Function to withdraw amount

**public** **void** withdraw(**int** x) {

balance=balance-x;

System.***out***.println("The amount withdrawn: "+getBalance());

}

//Inner class

**private** **class** InterestAdder{

//Function to calculate the interest on current balance

**void** displayBalance(**int** m,**double** b)

{

**double** balance1=b+(m\*(b\*0.12)/12);

System.***out***.println("The updated balance is: "+balance1);

}

}

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

{

//Outer class object

BankAccount b1=**new** BankAccount();

//Inner class object

BankAccount.InterestAdder i = b1.**new** InterestAdder();

b1.setBalance(30000);

**double** b2= b1.getBalance();

b1.deposit(10000);

b1.withdraw(5000);

i.displayBalance(42,b2);

}

}

**Output:**

The changed amount after deposit is: 40000.0

The amount withdrawn: 35000.0

The updated balance is: 42600.0

**Exercise 48:** *Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create a local inner class inside the getBalance() method called InterestAdder and implement*

*the interest calculations and add the interest to the current balance.*

**Solution:**

**package** com.hsbc.day3;

/\*Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.

Create a local inner class inside the getBalance() method called InterestAdder and implement

the interest calculations and add the interest to the current balance.

\*/

//outer class

**public** **class** BankAcoount1{

**private** **double** balance;

**public** **double** getBalance() {

**return** balance;

}

**public** **void** calculate(**int** m)

{

//Local inner class

**class** InterestAdder{

//Function to calculate the interest on current balance

**void** displayBalance(**int** m)

{

**double** balance1=balance+(m\*(balance\*0.12)/12);

System.***out***.println("The updated balance is: "+balance1);

}

}

InterestAdder i=**new** InterestAdder();

i.displayBalance(m);

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

//Function to deposit amount

**public** **void** deposit(**int** x) {

balance=balance+x;

System.***out***.println("The changed amount after deposit is: "+ getBalance());

}

//Function to withdraw amount

**public** **void** withdraw(**int** x) {

balance=balance-x;

System.***out***.println("The amount withdrawn: "+getBalance());

}

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

{

//Outer class object

BankAcoount1 b1=**new** BankAcoount1();

b1.setBalance(30000);

b1.deposit(10000);

b1.withdraw(5000);

b1.calculate(42);;

}

}

**Output:**

The changed amount after deposit is: 40000.0

The amount withdrawn: 35000.0

The updated balance is: 49700.0

**Exercise 49:** *Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.*

*Create an anonymous inner class in the getBalance() method to do the interest calculations*

*and add the interest to the current balance.*

**Solution:**

**package** com.hsbc.day3;

/\*Create a class called BankAccount with deposit (), withdraw () and getBalance () methods.

Create an anonymous inner class in the getBalance() method to do the interest calculations

and add the interest to the current balance.

\*/

//Interface declaration

**interface** InterestAdder{

**public** **void** displayBalance(**int** m);

}

//outer class

**public** **class** BankAccount3{

**private** **double** balance;

**public** **double** getBalance() {

**return** balance;

}

**public** **void** calculate(**int** m)

{

//Anonymous class

InterestAdder i=**new** InterestAdder(){

//Function to calculate the interest on current balance

**public** **void** displayBalance(**int** m)

{

**double** balance1=balance+(m\*(balance\*0.12)/12);

System.***out***.println("The updated balance is: "+balance1);

}

};

i.displayBalance(m);

}

**public** **void** setBalance(**double** balance) {

**this**.balance = balance;

}

//Function to deposit amount

**public** **void** deposit(**int** x) {

balance=balance+x;

System.***out***.println("The changed amount after deposit is: "+ getBalance());

}

//Function to withdraw amount

**public** **void** withdraw(**int** x) {

balance=balance-x;

System.***out***.println("The amount withdrawn: "+getBalance());

}

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

{

//Outer class object

BankAccount3 b1=**new** BankAccount3();

b1.setBalance(30000);

b1.deposit(10000);

b1.withdraw(5000);

b1.calculate(42);;

}

}

**Output:**

The changed amount after deposit is: 40000.0

The amount withdrawn: 35000.0

The updated balance is: 49700.0