**Name: Kunal Porwal**

Lab 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.

\*/

package com.hsbc.exercise34;

public class Vehicle {

public Vehicle() { //Default constructor

super();

}

}

//Subclasses which extends Vehicle class

class Truck extends Vehicle{

public Truck() {

super();

}

public void printVeh() {

System.out.println("In Truck.");

}

}

class Bus extends Vehicle{

public Bus() {

super();

}

public void printVeh() {

System.out.println("In Bus.");

}

}

class Car extends Vehicle{

public Car() {

super();

}

public void printVeh() {

System.out.println("In Car.");

}

}

class Road{

public static void main(String[] args) {

Truck t = new Truck();

Bus b = new Bus();

Car c = new Car();

t.printVeh();

b.printVeh();

c.printVeh();

}

}

Lab 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.

\*/

package com.hsbc.exercise35;

public class Vehicle {

//Variables

String color;

int noOfVehicles;

String model;

public Vehicle() {

super();

}

//Parameterised constructor

public Vehicle(String color, int noOfVehicles, String model) {

super();

this.color = color;

this.noOfVehicles = noOfVehicles;

this.model = model;

}

}

//Class truck inherited from vehicle

class Truck extends Vehicle {

public Truck() { //Default constructor

super("Red",4520,"T45");

}

public void printDetails() { //Print details method

System.out.println("Vehicle color:"+this.color+"\tNo of vehicles:"+this.noOfVehicles+"\tModel:"+this.model);

}

}

//Class car inherited from vehicle

class Car extends Vehicle {

public Car() { //Default constructor

super("Indigo",420,"A8");

}

public void printDetails() { //Print details method

System.out.println("Vehicle color:"+this.color+"\tNo of vehicles:"+this.noOfVehicles+"\tModel:"+this.model);

}

}

//Class Bus inherited from vehicle

class Bus extends Vehicle {

public Bus() { //Default constructor

super("Blue",120,"Bharat benz B56s");

}

public void printDetails() { //Print details method

System.out.println("Vehicle color:"+this.color+"\tNo of vehicles:"+this.noOfVehicles+"\tModel:"+this.model);

}

}

class Road { //Caller class

public static void main(String[] args) { //main method

Truck t = new Truck();

Bus b = new Bus();

Car c = new Car();

t.printDetails();

b.printDetails();

c.printDetails();

}

}

Lab exercise 36:

/\*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.

\*/

package com.hsbc.exercise36;

public class Vehicle {

//Variables

protected String color;

protected int noOfVehicles;

protected String model;

public Vehicle() {

super();

}

//Parameterised constructor

public Vehicle(String color, int noOfVehicles, String model) {

super();

this.color = color;

this.noOfVehicles = noOfVehicles;

this.model = model;

}

protected String getColor() {

return color;

}

protected void setColor(String color) {

this.color = color;

}

protected int getNoOfVehicles() {

return noOfVehicles;

}

protected void setNoOfVehicles(int noOfVehicles) {

this.noOfVehicles = noOfVehicles;

}

protected String getModel() {

return model;

}

protected void setModel(String model) {

this.model = model;

}

protected void display(Car c, Truck t, Bus b) {

System.out.println(c+"\n"+t+"\n"+b);

}

}

//Class truck inherited from vehicle

class Truck extends Vehicle {

protected Truck() { //Default constructor

super("Red",4520,"T45");

}

//toString method to display object in string format

@Override

public String toString() {

return "Truck [color=" + color + ", noOfVehicles=" + noOfVehicles + ", model=" + model + "]";

}

}

//Class car inherited from vehicle

class Car extends Vehicle {

protected Car() { //Default constructor

super("Indigo",420,"A8");

}

//toString method to display object in string format

@Override

public String toString() {

return "Car [color=" + color + ", noOfVehicles=" + noOfVehicles + ", model=" + model + "]";

}

}

//Class Bus inherited from vehicle

class Bus extends Vehicle {

protected Bus() { //Default constructor

super("Blue",120,"Bharat benz B56s");

}

//toString method to display object in string format

@Override

public String toString() {

return "Bus [color=" + color + ", noOfVehicles=" + noOfVehicles + ", model=" + model + "]";

}

}

/\*

\* class Road { //Caller class

\*

\* public static void main(String[] args) { //main method Truck t = new Truck();

\* Bus b = new Bus(); Car c = new Car();

\*

\* t.printDetails(); b.printDetails(); c.printDetails(); } }

\*/

//City class as caller class

class City extends Vehicle{

//main method

public static void main(String[] args) {

//objects of classes

Car c = new Car();

Truck t = new Truck();

Bus b = new Bus();

City city = new City();

city.display(c,t,b);

}

}

Lab exercise 36:

/\*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.

\*/

package com.hsbc.exercise39;

public class Worker { //base class

String name;

int salaryrate;

int noOfHours;

public Worker() {

super();

// TODO Auto-generated constructor stub

}

public Worker(String name, int salaryrate, int noOfHours) {

super();

this.name = name;

this.salaryrate = salaryrate;

this.noOfHours = noOfHours;

}

public int getNoOfHours() {

return noOfHours;

}

public void setNoOfHours(int noOfHours) {

this.noOfHours = noOfHours;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getSalaryrate() {

return salaryrate;

}

public void setSalaryrate(int salaryrate) {

this.salaryrate = salaryrate;

}

}

//Subclasses of worker class

class DailyWorker extends Worker{

public DailyWorker() {

super();

// TODO Auto-generated constructor stub

}

public DailyWorker(String name, int salaryrate, int noOfHours) {

super(name, salaryrate, noOfHours);

// TODO Auto-generated constructor stub

} //Inherited class

public int Pay(int noOfHours,int salarayRate) {

return noOfHours\*salarayRate;

}

}

class SalariedWorker extends Worker{

public SalariedWorker(String string, int i, int j) {

// TODO Auto-generated constructor stub

} //Inherited class

public int Pay(int noOfHours,int salarayRate) {

return 40\*salarayRate;

}

}

//Caller main method

class CalculatePay{

public static void main(String[] args) {

DailyWorker dw = new DailyWorker("Kunal Porwal",210,9);

SalariedWorker sw = new SalariedWorker("Om Shanti",200,12);

int salary=dw.Pay(dw.getNoOfHours(), dw.getSalaryrate());

System.out.println("Salary given to Daily Worker is:"+salary);

salary=sw.Pay(dw.getNoOfHours(), dw.getSalaryrate());

System.out.println("Salary given Salaried Worker is:"+salary);

}

}

Lab 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.\*/

package com.hsbc.exercise41;

abstract public class Shape3D {

//Abstract methods

abstract public void volume();

abstract public void surfaceArea();

final float pi = 3.14f;

}

//Subclasses

class Cylinder extends Shape3D{

//variables

double r=7.12,h=11.46;

double volume;

double surfaceArea;

@Override

public void volume() {

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

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

}

@Override

public void surfaceArea() {

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

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

}

}

class Sphere extends Shape3D{

//variables

double r=5.67;

double volume;

double surfaceArea;

@Override

public void volume() {

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

System.out.println("Volume of Sphere is:"+volume);

}

@Override

public void surfaceArea() {

// TODO Auto-generated method stub

surfaceArea = 4 \* pi \* r \* r;

System.out.println("Surface area of Sphere is:"+surfaceArea);

}

}

class Cube extends Shape3D{

//variables

double r=4.12;

double volume;

double surfaceArea;

@Override

public void volume() {

volume = r \* r \* r;

System.out.println("Volume of Cube is:"+volume);

}

@Override

public void surfaceArea() {

// TODO Auto-generated method stub

surfaceArea = 6 \* r \* r;

System.out.println("Surface area of Cube is:"+surfaceArea);

}

}

//Caller class

class mainMethod{

//main method

public static void main(String[] args) {

Cylinder cyl = new Cylinder();

cyl.volume();

cyl.surfaceArea();

Sphere sp = new Sphere();

sp.volume();

sp.surfaceArea();

Cube cb = new Cube();

cb.volume();

cb.surfaceArea();

}

}

Lab exercise 43:

/\*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.\*/

package com.hsbc.exercise43;

public interface Shape3D {

// methods from interface

public void volume();

public void surfaceArea();

final float pi = 3.14f;

}

//Subclasses

class Cylinder implements Shape3D{

//variables

double r=7.12,h=11.46;

double volume;

double surfaceArea;

@Override

public void volume() {

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

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

}

@Override

public void surfaceArea() {

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

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

}

}

class Sphere implements Shape3D{

//variables

double r=5.67;

double volume;

double surfaceArea;

@Override

public void volume() {

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

System.out.println("Volume of Sphere is:"+volume);

}

@Override

public void surfaceArea() {

// TODO Auto-generated method stub

surfaceArea = 4 \* pi \* r \* r;

System.out.println("Surface area of Sphere is:"+surfaceArea);

}

}

class Cube implements Shape3D{

//variables

double r=4.12;

double volume;

double surfaceArea;

@Override

public void volume() {

volume = r \* r \* r;

System.out.println("Volume of Cube is:"+volume);

}

@Override

public void surfaceArea() {

// TODO Auto-generated method stub

surfaceArea = 6 \* r \* r;

System.out.println("Surface area of Cube is:"+surfaceArea);

}

}

//Caller class

class mainMethod{

//main method

public static void main(String[] args) {

Cylinder cyl = new Cylinder();

cyl.volume();

cyl.surfaceArea();

Sphere sp = new Sphere();

sp.volume();

sp.surfaceArea();

Cube cb = new Cube();

cb.volume();

cb.surfaceArea();

}

}

Lab exercise 44:

**package** com.hsbc.exercise44;

//Interface

**public** **interface** Drawable {

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

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

}

//Interface

**interface** Fillable{

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

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

}

//Classes implements above interfaces

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

//Implemented methods

@Override

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

System.***out***.println("Filling color in Line");

}

@Override

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

System.***out***.println("In size of Line method");

}

@Override

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

System.***out***.println("Drawing color in Line");

}

@Override

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

System.***out***.println("Thickness of line method\n");

}

}

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

//Implemented methods

@Override

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

System.***out***.println("Filling color in Circle");

}

@Override

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

System.***out***.println("In size of Circle method");

}

@Override

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

System.***out***.println("Drawing color in Circle");

}

@Override

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

System.***out***.println("Thickness of Circle method\n");

}

}

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

//Implemented methods

@Override

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

System.***out***.println("Filling color in Square");

}

@Override

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

System.***out***.println("In size of Square method");

}

@Override

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

System.***out***.println("Drawing color in Square");

}

@Override

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

System.***out***.println("Thickness of Square method\n");

}

}

//Caller class

**class** Caller{

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

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

l.fillingColor();

l.size();

l.drawingColor();

l.thickness();

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

c.fillingColor();

c.size();

c.drawingColor();

c.thickness();

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

s.fillingColor();

s.size();

s.drawingColor();

s.thickness();

}

}

Lab exercise 44:

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

Payable.It should define the getPayInfo() methodthat all the worker classeswillimplement.

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

\*/

package com.hsbc.exercise45;

import com.hsbc.exercise45.finance.Payable;

public class Worker { //base class

String name;

int salaryrate;

int noOfHours;

public Worker() {

super();

// TODO Auto-generated constructor stub

}

public Worker(String name, int salaryrate, int noOfHours) {

super();

this.name = name;

this.salaryrate = salaryrate;

this.noOfHours = noOfHours;

}

public int getNoOfHours() {

return noOfHours;

}

public void setNoOfHours(int noOfHours) {

this.noOfHours = noOfHours;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getSalaryrate() {

return salaryrate;

}

public void setSalaryrate(int salaryrate) {

this.salaryrate = salaryrate;

}

}

//Subclasses of worker class

class DailyWorker extends Worker implements Payable{

public DailyWorker() {

super();

// TODO Auto-generated constructor stub

}

public DailyWorker(String name, int salaryrate, int noOfHours) {

super(name, salaryrate, noOfHours);

// TODO Auto-generated constructor stub

}

//Implemented method

@Override

public int getPayInfo() {

return this.getNoOfHours() \* this.getSalaryrate();

}

}

class SalariedWorker extends Worker implements Payable{

public SalariedWorker(String name, int salaryrate, int noOfHours) {

super(name, salaryrate, noOfHours);

}

//Implemented method

@Override

public int getPayInfo() {

return 40 \* this.getSalaryrate();

}

}

//Caller main method

class CalculatePay{

public static void main(String[] args) {

DailyWorker dw = new DailyWorker("Kunal Porwal",210,9);

SalariedWorker sw = new SalariedWorker("Om Shanti",200,12);

int salary=dw.getPayInfo();

System.out.println("Salary given to Daily Worker "+dw.getName()+" is:"+salary);

salary=sw.getPayInfo();

System.out.println("Salary given Salaried Worker "+sw.getName()+" is:"+salary);

dw.getPayInfo();

}

}

Lab exercise 45:

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

Payable.It should define the getPayInfo() methodthat all the worker classeswillimplement.

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

\*/

package com.hsbc.exercise45;

import com.hsbc.exercise45.finance.Payable;

public class Worker { //base class

String name;

int salaryrate;

int noOfHours;

public Worker() {

super();

// TODO Auto-generated constructor stub

}

public Worker(String name, int salaryrate, int noOfHours) {

super();

this.name = name;

this.salaryrate = salaryrate;

this.noOfHours = noOfHours;

}

public int getNoOfHours() {

return noOfHours;

}

public void setNoOfHours(int noOfHours) {

this.noOfHours = noOfHours;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public int getSalaryrate() {

return salaryrate;

}

public void setSalaryrate(int salaryrate) {

this.salaryrate = salaryrate;

}

}

//Subclasses of worker class

class DailyWorker extends Worker implements Payable{

public DailyWorker() {

super();

// TODO Auto-generated constructor stub

}

public DailyWorker(String name, int salaryrate, int noOfHours) {

super(name, salaryrate, noOfHours);

// TODO Auto-generated constructor stub

}

//Implemented method

@Override

public int getPayInfo() {

return this.getNoOfHours() \* this.getSalaryrate();

}

}

class SalariedWorker extends Worker implements Payable{

public SalariedWorker(String name, int salaryrate, int noOfHours) {

super(name, salaryrate, noOfHours);

}

//Implemented method

@Override

public int getPayInfo() {

return 40 \* this.getSalaryrate();

}

}

//Caller main method

class CalculatePay{

public static void main(String[] args) {

DailyWorker dw = new DailyWorker("Kunal Porwal",210,9);

SalariedWorker sw = new SalariedWorker("Om Shanti",200,12);

int salary=dw.getPayInfo();

System.out.println("Salary given to Daily Worker "+dw.getName()+" is:"+salary);

salary=sw.getPayInfo();

System.out.println("Salary given Salaried Worker "+sw.getName()+" is:"+salary);

dw.getPayInfo();

}

}

**package** com.hsbc.exercise45.finance;

**public** **interface** Payable {

**int** getPayInfo();

}

Lab exercise 46:

Account.java

**package** com.hsbc.exercise46.bank;

**public** **interface** Account {

**public** **static** **final** String ***Savings*** = "";

**public** **static** **final** String ***Fixed*** = "";

**public** **static** **final** String ***PersonalLoan*** = "";

**public** **static** **final** String ***HousingLoan*** = "";

**public** **void** createAcc();

}

CreditInterest.java

**package** com.hsbc.exercise46.bank;

**public** **interface** CreditInterest {

**public** **double** addMonthlyInt();

**public** **double** addHalfYrlyInt();

**public** **double** addAnnualInt();

}

DebitInterest.java

**package** com.hsbc.exercise46.bank;

**public** **interface** DebitInterest {

**public** **double** deductMonthlyInt();

**public** **double** deductHalfYrlyInt();

**public** **double** deductAnnualInt();

}

DepositAcc.java

**package** com.hsbc.exercise46.bank;

**public** **interface** DepositAcc {

**public** **double** withdraw ();

**public** **double** deposit();

**public** **double** getBalance();

}

Interest.java

**package** com.hsbc.exercise46.bank;

**public** **interface** Interest {

**public** **static** **final** **double** ***intPercentageSavingAcc*** = 0;

**public** **static** **final** **double** ***intPercentageFixedAcc*** = 0;

**public** **static** **final** **double** ***intPercentagePersonalLoanAcc*** = 0;

**public** **static** **final** **double** ***intPercentageHousingLoanAcc*** = 0;

**public** **void** calcInt();

}

LoanAcc.java

**package** com.hsbc.exercise46.bank;

**public** **interface** LoanAcc {

**public** **double** repayPrincipal ();

**public** **double** payInterest ();

**public** **double** payPartialPrincipal ();

}

Lab exercise 47:

**package** com.hsbc.exercise47;

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

//variables

**static** **double** *totalBalance*;

**double** deposit;

**double** withdraw;

**public** BankAccount() {

**super**();

// **TODO** Auto-generated constructor stub

}

**public** **double** getDeposit() {

**return** deposit;

}

**public** **void** setDeposit(**double** deposit) {

**this**.deposit = deposit;

}

**public** **double** getWithdraw() {

**return** withdraw;

}

**public** **void** setWithdraw(**double** withdraw) {

**this**.withdraw = withdraw;

}

**public** **double** getTotalBalance() {

**return** *totalBalance*;

}

**public** **void** setTotalBalance(**double** totalBalance) {

BankAccount.*totalBalance* = totalBalance;

}

**public** **double** deposit() {

**return** **this**.deposit;

}

**public** **double** withdraw(){

**return** **this**.withdraw;

}

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

**return** **this**.getTotalBalance();

}

//Inner class

**class** InterestAdder {

**public** **double** calcInt() {

**double** balance=BankAccount.*totalBalance*;

**return** balance \* 0.12 / 12;

}

}

}

//Caller class

**class** Caller{

//Main method

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

BankAccount b = **new** BankAccount();

b.setTotalBalance(2500.45);

b.setDeposit(2000.13);

b.setWithdraw(454.46);

BankAccount.InterestAdder in = **new** BankAccount().**new** InterestAdder();

System.***out***.println("Interest for a month:"+in.calcInt());

}

}

Lab 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.

\*/

package com.hsbc.exercise48;

public class BankAccount {

//variables

static double totalBalance;

double deposit;

double withdraw;

public BankAccount() {

super();

// TODO Auto-generated constructor stub

}

public double getDeposit() {

return deposit;

}

public void setDeposit(double deposit) {

this.deposit = deposit;

}

public double getWithdraw() {

return withdraw;

}

public void setWithdraw(double withdraw) {

this.withdraw = withdraw;

}

public double getTotalBalance() {

return totalBalance;

}

public void setTotalBalance(double totalBalance) {

BankAccount.totalBalance = totalBalance;

}

public double deposit() {

return this.deposit;

}

public double withdraw(){

return this.withdraw;

}

public double getBalance() {

//Local inner class

class InterestAdder {

public double calcInt() {

double balance=BankAccount.totalBalance;

return balance \* 0.12 / 12;

}

}

InterestAdder i = new InterestAdder();

return i.calcInt();

}

}

//Caller class

class Caller{

//main method

public static void main(String[] args) {

BankAccount b = new BankAccount();

b.setTotalBalance(2500.45);

b.setDeposit(2000.13);

b.setWithdraw(454.46);

//BankAccount.InterestAdder in = new BankAccount().new InterestAdder();

System.out.println("Interest for a month:"+b.getBalance());

}

}

Lab 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.

\*/

package com.hsbc.exercise49;

public class BankAccount {

//variables

static double totalBalance;

double deposit;

double withdraw;

void deposit(int x) {

totalBalance = totalBalance+x;

}

void withdraw(int x) {

totalBalance = totalBalance-x;

}

double getBalance() {

Calculate p=new Calculate(){

double interest(){

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

}

};

totalBalance=totalBalance+p.interest();

return totalBalance;

}

public static void main(String[] args)

{

BankAccount b=new BankAccount();

b.deposit(12987);

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

}

}

abstract class Calculate{

abstract double interest();

}