**abstract**

**Exercise 1:***Create a class called Shape3D with the following method signatures alone, volume () and*

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

*these methods.*

**Inheritance**

**Exercise 2:***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.*

**Exercise 3:***In the Lab Exercise above, 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.*

**Exercise 4:***In the Lab Exercise above, 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.*

package com.vehicle;

public class Vehicle {

private String color,model;

private int noOfWeels;

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

super();

this.color = color;

this.model = model;

this.noOfWeels = noOfWeels;

}

void start() {

System.out.println("Vehicle started");

}

void breaks()

{

System.out.println("Vehicle stoped");

}

public void display() {

System.out.println("");

}

}

package com.vehicle;

public class Truck extends Vehicle {

private String color,model;

private int noOfWeels;

public Truck(String color, String model, int noOfWeels) {

super(color, model, noOfWeels);

this.color=color;

this.model=model;

this.noOfWeels=noOfWeels;

}

@Override

public void start() {

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

}

public String getColor() {

return color;

}

public String getModel() {

return model;

}

public int getNoOfWeels() {

return noOfWeels;

}

@Override

public void breaks() {

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

}

public void loadingCapacity() {

System.out.println("Truck can load 50 tuns");

}

public void display() {

System.out.println("Truck color is:"+getColor()+" Truck model is:"+getModel()+" No of Weels are:"+getNoOfWeels());

}

}

package com.vehicle;

public class Bus extends Vehicle {

private String color,model;

private int noOfWeels;

public String getColor() {

return color;

}

public String getModel() {

return model;

}

public int getNoOfWeels() {

return noOfWeels;

}

public Bus(String color, String model, int noOfWeels) {

super(color, model, noOfWeels);

this.color=color;

this.model=model;

this.noOfWeels=noOfWeels;

}

@Override

public void start()

{

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

}

@Override

public void breaks() {

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

}

public void carries() {

System.out.println("Bus carries people");

}

public void display() {

System.out.println("Bus color is:"+getColor()+" Bus model is:"+getModel()+" No of Weels are:"+getNoOfWeels());

}

}

package com.vehicle;

public class Car extends Vehicle {

private String color,model;

private int noOfWeels;

public String getColor() {

return color;

}

public String getModel() {

return model;

}

public int getNoOfWeels() {

return noOfWeels;

}

public Car(String color, String model, int noOfWeels) {

super(color, model, noOfWeels);

this.color=color;

this.model=model;

this.noOfWeels=noOfWeels;

}

@Override

public void start() {

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

}

@Override

public void breaks() {

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

}

public void speed()

{

System.out.println("Car can goes with 300 kmph");

}

public void display() {

System.out.println("Car color is: "+getColor()+"Car model is:"+getModel()+" No of Weels are:"+getNoOfWeels());

}

}

package com.vehicle;

public class Road {

public static void main(String[] args) {

Vehicle t=new Truck("red", "Truck 101", 6);

t.start();

t.breaks();

((Truck)t).loadingCapacity();

Vehicle b=new Bus("yellow","Bus 101",6);

b.start();

b.breaks();

((Bus)b).carries();

Vehicle c=new Car("White","Car 101",4);

c.start();

c.breaks();

((Car)c).speed();

}

}

package com.vehicle;

public class City {

public static void main(String[] args) {

Vehicle t=new Truck("red", "Truck 101", 6);

t.display();

t=new Bus("yellow","Bus 101",6);

t.display();

t=new Car("White","Car 101",4);

t.display();

}

}

**Polymorphism**

**Exercise 5:***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.*

**Interfaces**

**Exercise 6:***Createa package called bank with the following Interfaces.*

*<Interface> Account*

*<Interface>DepositAcc <Interface>LoanAcc*

*<Interface> Interest*

*<Interface>CreditInterest <Interface>DebitInterest*

1. *<Interface> Account*

*Data members: Four String variables to hold the account type “Savings, Fixed,PersonalLoan,*

*HousingLoan”*

*Methods: createAcc()*

1. *<Interface>DepositAcc*

*Methods: withdraw (), deposit(),getBalance()*

1. *<Interface>LoanAcc*

*Methods: repayPrincipal (),payInterest (),payPartialPrincipal ()*

1. *<Interface>Interest*

*Data members: Four double variables to hold the interest percentage of Savings account, Fixed*

*deposit account,PersonalLoan account and HousingLoan account.*

*Methods: calcInt()*

1. *<Interface>CreditInterest*

*Methods: addMonthlyInt(),addHalfYrlyInt(),addAnnualInt()*

1. *<Interface>DebitInterest*

*Methods: deductMonthlyInt(),deductHalfYrlyInt(),deductAnnualInt()*

*Create a package called BankImpl and create the following classes in it.*

1. *SavingsAcc which implements DepositAcc and CreditInterest*
2. *FDAcc which implements DepositAcc and CreditInterest*
3. *PersonalLoanAcc which implements LoanAcc and DebitInterest*
4. *HousingLoanAcc which implements LoanAcc and DebitInterest*

*Now create a class called MyAccount and create instances of all the accounts and generate appropriate output.*

**Static**

**Exercise 7:***Create a class called Sample. Write a program to display the no of objects created for that*

*class or the no of times that class is instantiated.*