****

**LAB-8**

By-Arun Lal

****

Sec-E

BSCS(II)

CMS-ID=023-24-0120

**Exercise of Lab: 8**

**Question no: 01 - Part(a)**

**public class Employee{**

**String firstName ;**

**String lastName ;**

**String CNIC ;**

**Employee(){}**

**Employee(String firstName,String lastName,String CNIC){**

**this.firstName = firstName ;**

**this.lastName = lastName ;**

**this.CNIC = CNIC ;**

**}**

**public String toString(){**

**return firstName + " " + lastName + " CNIC# " + CNIC ;**

**}**

**protected void first\_nameSetter(String firstName){**

**this.firstName = firstName ;**

**}**

**protected void last\_nameSetter(String lastName){**

**this.lastName = lastName ;**

**}**

**protected void id\_numberSetter(String CNIC){**

**this.CNIC = CNIC ;**

**}**

**protected String first\_nameGetter(){**

**return firstName ;**

**}**

**protected String last\_nameGetter(){**

**return lastName ;**

**}**

**protected String id\_numberGetter(){**

**return CNIC ;**

**}**

**public double earnings(){**

**return 0.00 ;**

**}**

**}**

**import java.util.Scanner ;**

**class SalariedEmployee extends Employee{**

**double weeklySalary ;**

**SalariedEmployee(){}**

**SalariedEmployee(String firstName,String lastName,String CNIC,double weeklySalary){**

**super(firstName,lastName,CNIC) ;**

**this.weeklySalary = weeklySalary ;**

**}**

**protected double getter(){**

**return weeklySalary ;**

**}**

**protected void setter(double weeklySalary){**

**if(weeklySalary >= 0){**

**this.weeklySalary = weeklySalary ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter vaild salary: ") ;**

**weeklySalary = cin.nextDouble() ;**

**}while(weeklySalary < 0) ;**

**this.weeklySalary = weeklySalary ;**

**}**

**}**

**public String toString() {**

**return "\nSalaried employee: " + super.toString() ;**

**}**

**public double earnings(){**

**return weeklySalary ;**

**}**

**}**

**import java.util.Scanner ;**

**class HourlyEmployee extends Employee{**

**double wage ;**

**double hours ;**

**HourlyEmployee(){}**

**HourlyEmployee(String firstName,String lastName,String CNIC,double wage,double hours){**

**super(firstName,lastName,CNIC) ;**

**this.wage = wage ;**

**this.hours = hours ;**

**}**

**protected double wageGetter(){**

**return wage ;**

**}**

**protected double hoursGetter(){**

**return hours ;**

**}**

**protected void wageSetter(double wage){**

**if(wage >= 0){**

**this.wage = wage ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter valid wage: ") ;**

**wage = cin.nextDouble() ;**

**}while(wage < 0) ;**

**this.wage = wage ;**

**}**

**}**

**protected void hoursSetter(double hours){**

**if(hours >= 0){**

**this.hours = hours ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter valid hours: ") ;**

**hours = cin.nextDouble() ;**

**}while(hours < 0) ;**

**this.hours = hours ;**

**}**

**}**

**public String toString(){**

**return "\nHourly employee: " + super.toString() ;**

**}**

**public double earnings(){**

**if(hours <= 40){**

**return wage \* hours;**

**}**

**else{**

**return 40\*wage + (hours-40)\*wage\*1.5;**

**}**

**}**

**}**

**import java.util.Scanner ;**

**class CommissionEmployee extends Employee{**

**double grossSale ;**

**double commissionRate ;**

**CommissionEmployee(){}**

**CommissionEmployee(String firstName,String lastName,String CNIC,double grossSale,double commissionRate)**

**{**

**super(firstName,lastName,CNIC) ;**

**this.grossSale = grossSale ;**

**this.commissionRate = commissionRate ;**

**}**

**protected double grossSaleGetter(){**

**return grossSale ;**

**}**

**protected double commissionRateGetter(){**

**return commissionRate ;**

**}**

**protected void grossSaleSetter(double grossSale){**

**if(grossSale >= 0){**

**this.grossSale = grossSale ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter valid gross Sale: ") ;**

**grossSale = cin.nextDouble() ;**

**}while(grossSale < 0) ;**

**}**

**}**

**protected void commissionRateStter(double commissionRate){**

**if(commissionRate >= 0){**

**this.commissionRate = commissionRate ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter valid commission rate: ") ;**

**commissionRate = cin.nextDouble() ;**

**}while(commissionRate < 0) ;**

**}**

**}**

**public String toString(){**

**return " \nCommission employee: " + super.toString() ;**

**}**

**public double earnings(){**

**return grossSale \* commissionRate ;**

**}**

**}**

**import java.util.Scanner ;**

**class BasePlusCommissionEmployee extends CommissionEmployee{**

**double baseSalary ;**

**BasePlusCommissionEmployee(){}**

**BasePlusCommissionEmployee(String firstName,String lastName,String CNIC,double grossSale,**

**double commissionRate,double baseSalary){**

**super(firstName,lastName,CNIC,grossSale,commissionRate) ;**

**this.baseSalary = baseSalary ;**

**}**

**protected void setter(double baseSalary){**

**if(baseSalary >= 0){**

**this.baseSalary = baseSalary ;**

**}**

**else{**

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

**do{**

**System.out.println("Invalid input!Please enter valid base salary: ") ;**

**baseSalary = cin.nextDouble() ;**

**}while(baseSalary < 0) ;**

**}**

**}**

**protected double getter(){**

**return baseSalary ;**

**}**

**public String toString(){**

**return "\nBase plus Commission employee: " + super.toString() ;**

**}**

**public double earnings(){**

**return baseSalary + super.earnings() ;**

**}**

**}**

**Question no: 01:- Part(b)**

**public class PayRollSystemTest {**

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

**Employee firstEmployee = new SalariedEmployee("Muhammad" ,"Ali","111111111", 800.00 );**

**Employee secondEmployee = new CommissionEmployee("Tarwan" ,"Kumar",**

**"222-22-2222", 10000, 0.06 ) ;**

**Employee thirdEmployee = new BasePlusCommissionEmployee("Fabeeha",**

**"Fatima", "333-33-3333", 5000 , 0.04 , 300 ) ;**

**Employee fourthEmployee = new HourlyEmployee( "Hasnain" , "Ali", "444-444444" ,**

**16.75 , 40 ) ;**

**// polymorphism: calling toString() and earning() on Employee’s reference**

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

**System.out.println(firstEmployee.earnings());**

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

**System.out.println(secondEmployee.earnings());**

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

**// performing downcasting to access & raise base salary**

**BasePlusCommissionEmployee currentEmployee = (BasePlusCommissionEmployee)thirdEmployee;**

**double oldBaseSalary = currentEmployee.getter();**

**System.out.println( "old base salary: " + oldBaseSalary) ;**

**currentEmployee.setter(1.10 \* oldBaseSalary) ;**

**System.out.println("new base salary with 10% increase is:"+**

**currentEmployee.getter()) ;**

**System.out.println(thirdEmployee.earnings() );**

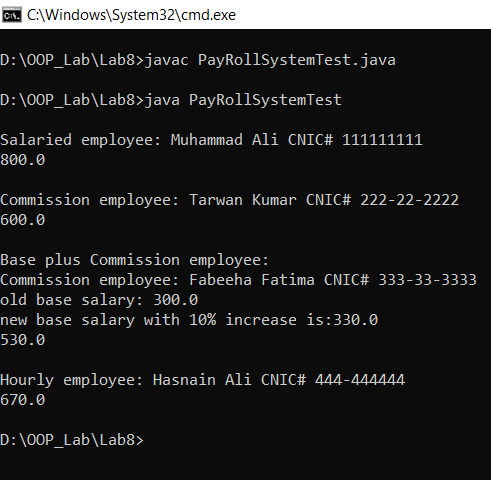
**System.out.println(fourthEmployee);**

**System.out.println(fourthEmployee.earnings() );**

**} // end main**

**} // end class**

**Output:**

****

**Question no: 02: (Part\_a)**

**Code:**

**class Point {**

**private int x ;**

**private int y ;**

**public Point(int x, int y){**

**this.x = x ;**

**this.y = y ;**

**}**

**public int getX(){**

**return x ;**

**}**

**public int getY(){**

**return y ;**

**}**

**public double distanceTo(Point p){**

**return Math.sqrt((x-p.getX())\*(x-p.getX()) + (y-p.getY())\*(y-p.getY())) ;**

**}**

**public String toString(){**

**return "("+x+", "+y+")" ;**

**}**

**}**

**class Shape{**

**private Point p ;**

**Shape(Point p){**

**this.p = p ;**

**}**

**public Point getCenter(){**

**return p ;**

**}**

**public boolean contains(Point p){**

**return false ;**

**}**

**}**

**class Circle extends Shape{**

**private int radius ;**

**Circle(int radius,Point c){**

**super(c) ;**

**this.radius = radius ;**

**}**

**public int getRadius(){**

**return radius ;**

**}**

**public String toString(){**

**return "Radius: " + radius + ", center(" + getCenter().getX() + "," +**

**getCenter().getY() + ")" ;**

**}**

**public boolean contains(Point p){**

**return getCenter().distanceTo(p) <= radius ;**

**}**

**}**

**class Rectangle extends Shape{**

**private int length ;**

**private int width ;**

**Rectangle(int l,int w,Point c){**

**super(c) ;**

**length = l ;**

**width = w ;**

**}**

**public int getLength(){**

**return length ;**

**}**

**public int getWidth(){**

**return width ;**

**}**

**public String toString(){**

**return "Length: " + length + ", width: " + width + ", center(" + getCenter().getX()**

**+ "," + getCenter().getY() + ")" ;**

**}**

**public boolean contains(Point p){**

**Point center = getCenter();**

**return Math.abs(p.getX() - center.getX()) <= width / 2 &&**

**Math.abs(p.getY() - center.getY()) <= length / 2;**

**}**

**}**

**class ShapeArray{**

**Shape[] shapeList ;**

**int nbShapes = 0 ;**

**ShapeArray(int size){**

**shapeList = new Shape[size] ;**

**}**

**boolean addShape(Shape s){**

**if(s != null){**

**shapeList[nbShapes++] = s ;**

**return true ;**

**}**

**return false ;**

**}**

**int getCircleCounter(){**

**int count = 0 ;**

**for(int i=0 ; i<nbShapes ; i++){**

**if(shapeList[i] != null && shapeList[i] instanceof Circle){**

**count++ ;**

**}**

**}**

**return count ;**

**}**

**double getAvgRectArea(){**

**int count = 0 ;**

**double area = 0 ;**

**for(int i=0 ; i<nbShapes ; i++){**

**if(shapeList[i] != null && shapeList[i] instanceof Rectangle){**

**Rectangle rect = (Rectangle)shapeList[i] ;**

**area += rect.getLength() \* rect.getWidth() ;**

**count++ ;**

**}**

**}**

**if(count == 0){**

**return 0 ;**

**}**

**return area/count ;**

**}**

**void removeAllRect(){**

**int j = 0 ;**

**for(int i=0 ; i<nbShapes ; i++){**

**if(shapeList[i] != null && !(shapeList[i] instanceof Rectangle)){**

**shapeList[j++] = shapeList[i] ;**

**}**

**}**

**nbShapes = j ;**

**}**

**public String toString(){**

**return "Number of shapes: " + nbShapes ;**

**}**

**void displayrectsinfo(){**

**for(int i=0 ; i<nbShapes ; i++){**

**if(shapeList[i] != null && shapeList[i] instanceof Rectangle){**

**System.out.println(shapeList[i]) ;**

**}**

**}**

**}**

**}**

**Question no: 02: (Part\_b)**

**Code:**

**import java.util.Scanner;**

**public class TestShape {**

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

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

**ShapeArray shapeArray = new ShapeArray(20);**

**int choice;**

**do {**

**System.out.println("\nMenu:");**

**System.out.println("1. Add new shape");**

**System.out.println("2. Display all rectangles");**

**System.out.println("3. Display average rectangle area");**

**System.out.println("4. Display number of circles");**

**System.out.println("5. Remove all rectangles");**

**System.out.println("6. Exit");**

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

**choice = input.nextInt();**

**switch (choice) {**

**case 1:**

**System.out.println("a. Rectangle");**

**System.out.println("b. Circle");**

**System.out.print("Enter your choice (a/b): ");**

**char shapeType = input.next().toLowerCase().charAt(0);**

**if (shapeType == 'a') {**

**System.out.print("Enter length: ");**

**int length = input.nextInt();**

**System.out.print("Enter width: ");**

**int width = input.nextInt();**

**System.out.print("Enter center X: ");**

**int x = input.nextInt();**

**System.out.print("Enter center Y: ");**

**int y = input.nextInt();**

**Rectangle rect = new Rectangle(length, width, new Point(x, y));**

**if (shapeArray.addShape(rect)) {**

**System.out.println("Rectangle added.");**

**} else {**

**System.out.println("Shape array is full!");**

**}**

**} else if (shapeType == 'b') {**

**System.out.print("Enter radius: ");**

**int radius = input.nextInt();**

**System.out.print("Enter center X: ");**

**int x = input.nextInt();**

**System.out.print("Enter center Y: ");**

**int y = input.nextInt();**

**Circle circle = new Circle(radius, new Point(x, y));**

**if (shapeArray.addShape(circle)) {**

**System.out.println("Circle added.");**

**} else {**

**System.out.println("Shape array is full!");**

**}**

**} else {**

**System.out.println("Invalid shape choice.");**

**}**

**break;**

**case 2:**

**System.out.println("Rectangles:");**

**shapeArray.displayrectsinfo();**

**break;**

**case 3:**

**System.out.println("Average rectangle area: " + shapeArray.getAvgRectArea());**

**break;**

**case 4:**

**System.out.println("Number of circles: " + shapeArray.getCircleCounter());**

**break;**

**case 5:**

**shapeArray.removeAllRect();**

**System.out.println("All rectangles removed.");**

**break;**

**case 6:**

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

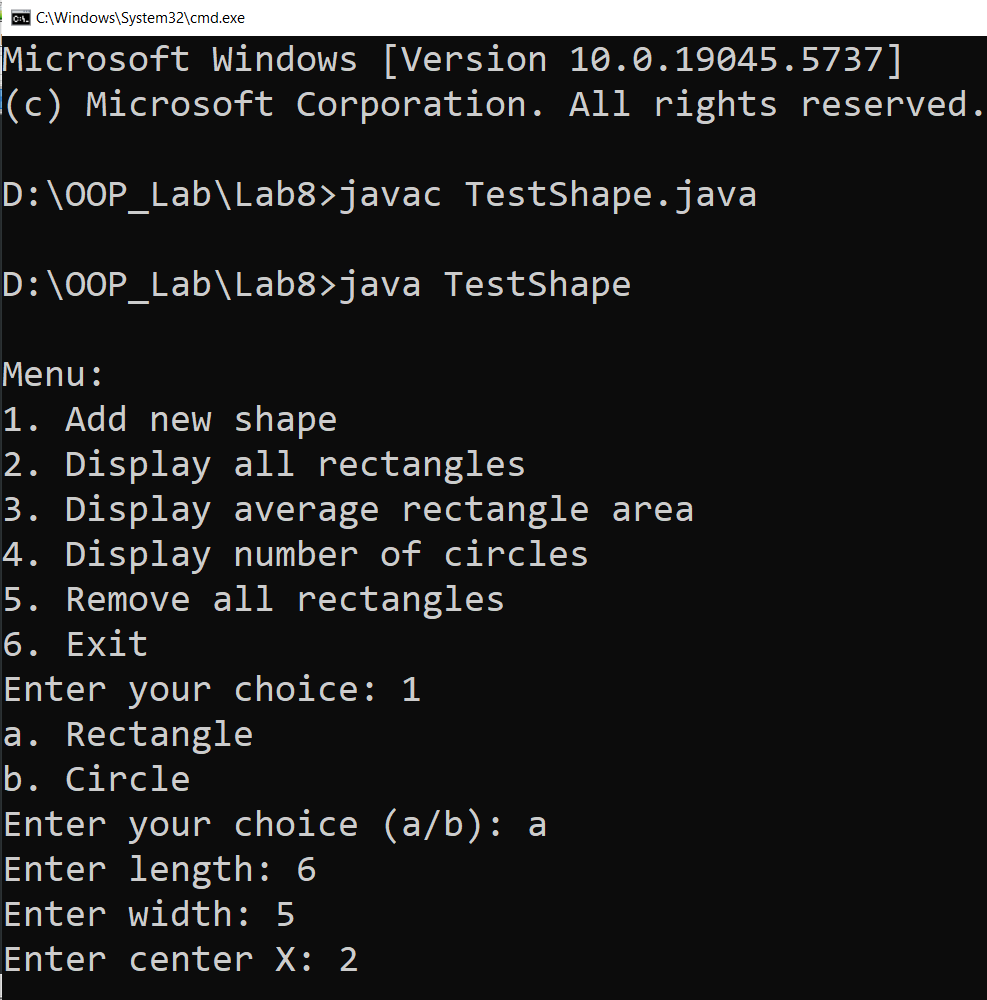
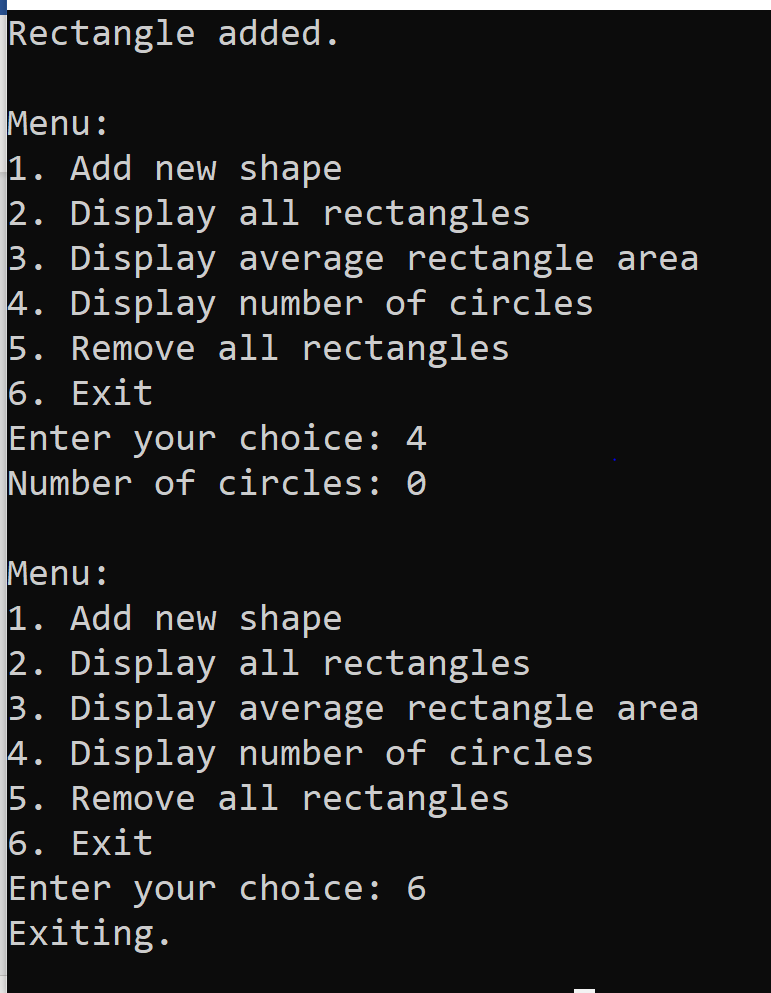
**break;**

**default:**

**System.out.println("Invalid choice. Please try again.");**

**}**

**} while (choice != 6) ;}}**

**Output:**

**Question no: 03:-**

**Code:**

**class Math{**

**void display(){**

**System.out.println("Hello I am display method of class Maths") ;**

**}**

**}**

**class Algebra extends Math{**

**void display(){**

**System.out.println("Hello I am display method of class Algebra") ;**

**}**

**}**

**class TestMath{**

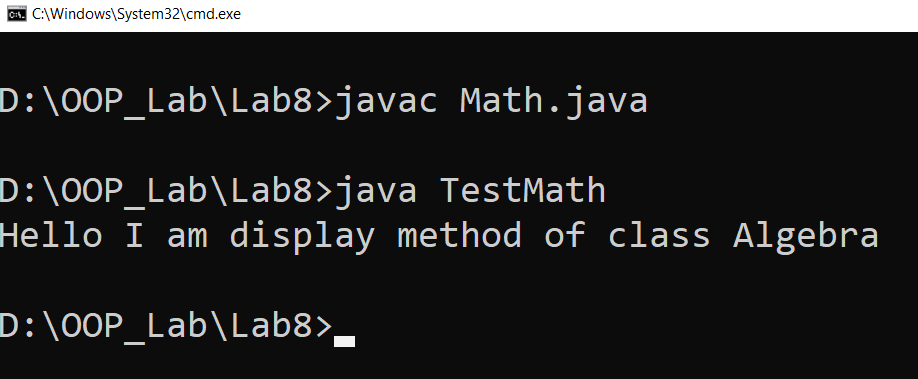
**public static void main(String[] args){**

**Math obj = new Algebra() ;**

**obj.display() ;**

**}**

**}**

**Output:**

**Question no: 04:-**

**Code:**

**public class Animal{**

**String name ;**

**int age ;**

**String gender ;**

**Animal(String name,int age,String gender){**

**this.name = name ;**

**this.age = age ;**

**this.gender = gender ;**

**}**

**void ProduceSound(){**

**System.out.println("Animal Sound") ;**

**}**

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

**Animal[] obj = new Animal[10] ;**

**obj[0] = new Dog("Spark",5,"male") ;**

**obj[1] = new Dog("Bob",6,"male") ;**

**obj[2] = new Dog("Doggy",4,"female") ;**

**obj[3] = new Kitten("Tomy",3) ;**

**obj[4] = new Kitten("Meow",2) ;**

**obj[5] = new Frog("Mendak",1,"male") ;**

**obj[6] = new Frog("Queen Mendaka",1,"female") ;**

**obj[7] = new Tomcat("Tom",12) ;**

**obj[8] = new Tomcat("Jack",10) ;**

**obj[9] = new Tomcat("Oggy",9) ;**

**double avgDog = 0 ;**

**double avgKitten = 0 ;**

**double avgFrog = 0 ;**

**double avgTomcat = 0 ;**

**for(int i = 0 ; i < 10 ; i++){**

**if(obj[i] != null){**

**if(obj[i] instanceof Dog){**

**avgDog += obj[i].age ;**

**}**

**else if(obj[i] instanceof Kitten){**

**avgKitten += obj[i].age ;**

**}**

**else if(obj[i] instanceof Tomcat){**

**avgFrog += obj[i].age ;**

**}**

**else{**

**avgTomcat += obj[i].age ;**

**}**

**}**

**}**

**System.out.println("Average age of dogs: " + avgDog) ;**

**System.out.println("Average age of frogs: " + avgFrog) ;**

**System.out.println("Average age of kitten: " + avgKitten) ;**

**System.out.println("Average age of tomcat: " + avgTomcat) ;**

**}**

**}**

**class Dog extends Animal{**

**Dog(String name,int age,String gender){**

**super(name,age,gender) ;**

**}**

**void ProduceSound(){**

**System.out.println("Bow bow") ;**

**}**

**void dogEatMeat(){**

**System.out.println(name + "'s special food is meat") ;**

**}**

**}**

**class Kitten extends Animal{**

**Kitten(String name,int age){**

**super(name,age,"female") ;**

**}**

**void ProduceSound(){**

**System.out.println("Kitten Meow") ;**

**}**

**void kittenAsMother(){**

**System.out.println(name + " can birth baby cats") ;**

**}**

**}**

**class Frog extends Animal{**

**Frog(String name,int age,String gender){**

**super(name,age,gender) ;**

**}**

**void ProduceSound(){**

**System.out.println("Tran tran") ;**

**}**

**void available(){**

**System.out.println(name + " found in small ponds") ;**

**}**

**}**

**class Tomcat extends Animal{**

**Tomcat(String name,int age){**

**super(name,age,"male") ;**

**}**

**void ProduceSound(){**

**System.out.println("Tomcat Meow") ;**

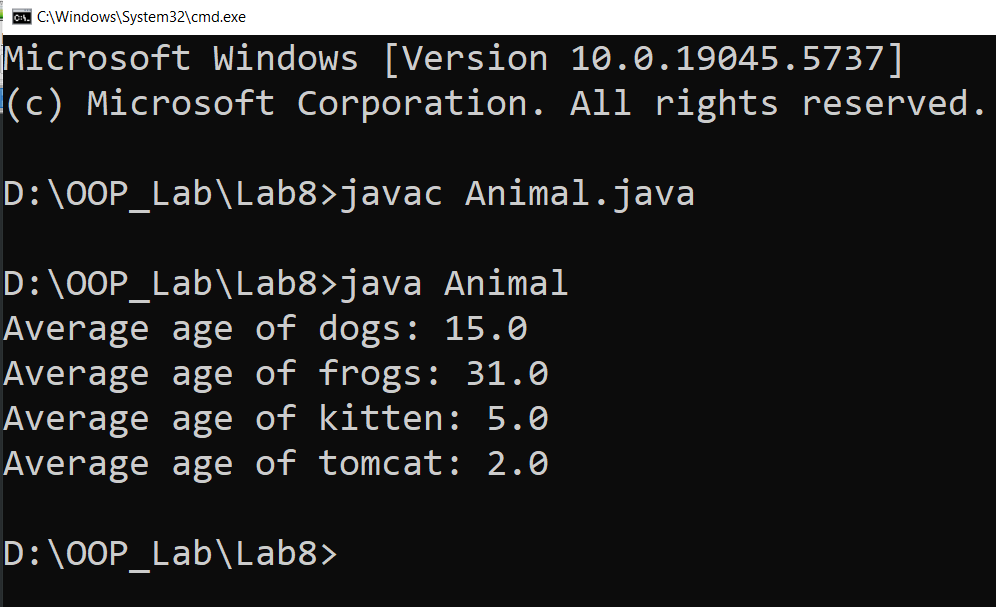
**}**

**void drinkMilk(){**

**System.out.println(name + " can drink milk") ;**

**}**

**}**

**Output:**

**Question no: 05:- (part a)**

**Code:**

**class Movie{**

**private int ID\_Number ;**

**private String title ;**

**private String mpaaRating ;**

**Movie(int ID\_Number,String title,String mpaaRating){**

**this.ID\_Number = ID\_Number ;**

**this.title = title ;**

**this.mpaaRating = mpaaRating ;**

**}**

**public boolean equals(Object obj) {**

**if (this == obj) return true;**

**if (obj == null || getClass() != obj.getClass()) return false;**

**Movie other = (Movie) obj;**

**return this.ID\_Number == other.ID\_Number;**

**} public void mpaaRatingSetter(String mpaaRating){**

**this.mpaaRating = mpaaRating ;**

**}**

**public String mpaaRatingGet(){**

**return mpaaRating ;**

**}**

**public void IDSetter(int id){**

**ID\_Number = id ;**

**}**

**public void titleSetter(String title){**

**this.title = title ;**

**}**

**public int IDGet(){**

**return ID\_Number ;**

**}**

**public String titleGet(){**

**return title ;**

**}**

**double calcLateFees(int day){**

**return day \* 2 ;**

**}**

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

**Movie[] obj = new Movie[3] ;**

**obj[0] = new Action(101,"Thor","Rated G") ;**

**obj[1] = new Comedy(102,"Total Dhamal","PG-13") ;**

**obj[2] = new Drama(103,"Pride and Prejudice","R") ;**

**System.out.println("Action movie ID: " + obj[0].IDGet() + "\nTitle: " +**

**obj[0].titleGet() + "\nMPAA Rating: " + obj[0].mpaaRatingGet()**

**+ "\n3 days late fee: " + obj[0].calcLateFees(3)) ;**

**System.out.println("\nComedy movie ID: " + obj[1].IDGet() + "\nTitle: " +**

**obj[1].titleGet() + "\nMPAA Rating: " + obj[1].mpaaRatingGet()**

**+ "\n4 days late fee: " + obj[1].calcLateFees(4)) ;**

**System.out.println("\nDrama ID: " + obj[2].IDGet() + "\nTitle: " +**

**obj[2].titleGet() + "\nMPAA Rating: " + obj[2].mpaaRatingGet()**

**+ "\n2 days late fee: " + obj[0].calcLateFees(2)) ;**

**}**

**}**

**class Action extends Movie{**

**Action(int id,String title,String mpaaRating){**

**super(id,title,mpaaRating) ;**

**}**

**double calcLateFees(int day){**

**return day \* 3 ;**

**}**

**}**

**class Comedy extends Movie{**

**Comedy(int id,String title,String mpaaRating){**

**super(id,title,mpaaRating) ;**

**}**

**double calcLateFees(int day){**

**return day \* 2.50 ;**

**}**

**}**

**class Drama extends Movie{**

**Drama(int id,String title,String mpaaRating){**

**super(id,title,mpaaRating) ;**

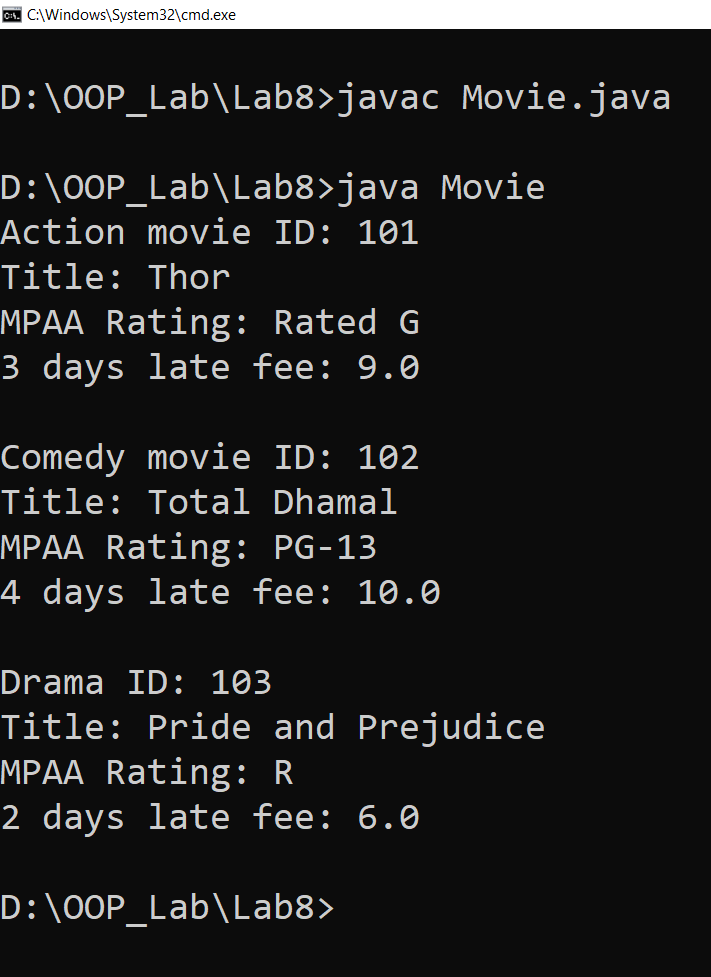
**}**

**double calcLateFees(int day){**

**return day \* 2 ;**

**}**

**}**

**Output:**

**Question no: 05:- (part b)**

**Code:**

**class Rental {**

**private Movie rentedMovie;**

**private int customerID;**

**private int daysLate;**

**Rental(Movie rentedMovie, int customerID, int daysLate) {**

**this.rentedMovie = rentedMovie;**

**this.customerID = customerID;**

**this.daysLate = daysLate;**

**}**

**public double calculateLateFee() {**

**return rentedMovie.calcLateFees(daysLate);**

**}**

**public Movie getRentedMovie() {**

**return rentedMovie;**

**}**

**public int getCustomerID() {**

**return customerID;**

**}**

**public int getDaysLate() {**

**return daysLate;**

**}**

**public static double lateFeesOwed(Rental[] rentals) {**

**double total = 0;**

**for (Rental r : rentals) {**

**total += r.calculateLateFee();**

**}**

**return total;**

**}**

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

**Rental[] rentals = new Rental[3];**

**rentals[0] = new Rental(new Action(101, "Thor", "Rated G"), 201, 3);**

**rentals[1] = new Rental(new Comedy(102, "Total Dhamal", "PG-13"), 202, 4);**

**rentals[2] = new Rental(new Drama(103, "Pride and Prejudice", "R"), 203, 2);**

**for (Rental r : rentals) {**

**System.out.println("Customer ID: " + r.getCustomerID());**

**System.out.println("Movie Title: " + r.getRentedMovie().titleGet());**

**System.out.println("Late by: " + r.getDaysLate() + " days");**

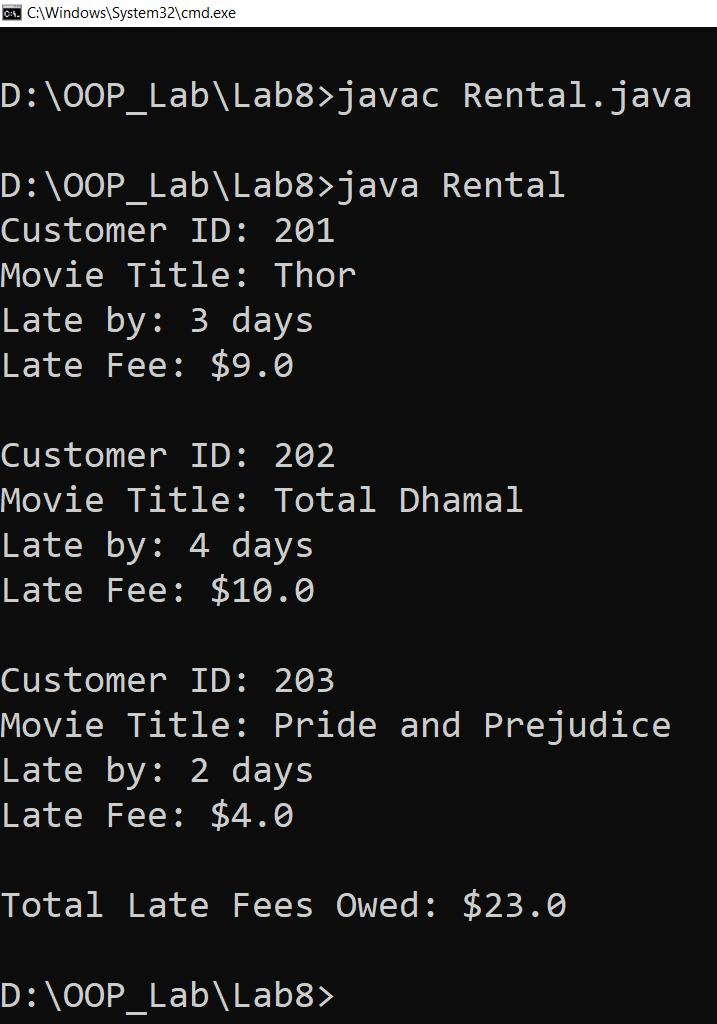
**System.out.println("Late Fee: $" + r.calculateLateFee() + "\n");**

**}**

**System.out.println("Total Late Fees Owed: $" + lateFeesOwed(rentals));**

**}**

**}**

****

**Question no: 06:-**

**Code:**

**package hospital.person ;**

**public class Person{**

**private String name ;**

**private String address ;**

**private String mobileNumber ;**

**private String ID ;**

**private String gender ;**

**private int age ;**

**public Person(String ID,String name,String gender,int age,String mobileNumber,**

**String address){**

**this.ID = ID ;**

**this.name = name ;**

**this.gender = gender ;**

**this.age = age ;**

**this.mobileNumber = mobileNumber ;**

**this.address = address ;**

**}**

**public Person(){} ;**

**public String nameGet(){**

**return name ;**

**}**

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

**this.name = name ;**

**}**

**public String addrGet(){**

**return address ;**

**}**

**public void addrSet(String address){**

**this.address = address ;**

**}**

**public String mobNumGet(){**

**return mobileNumber ;**

**}**

**public void mobNumSet(String mobileNumber){**

**this.mobileNumber = mobileNumber ;**

**}**

**public String IDGet(){**

**return ID ;**

**}**

**public void IDSet(String ID){**

**this.ID = ID ;**

**}**

**public String genderGet(){**

**return gender ;**

**}**

**public void genderSet(String gender){**

**this.gender = gender ;**

**}**

**public int ageGet(){**

**return age ;**

**}**

**public void ageSet(int age){**

**this.age = age ;**

**}**

**public void displayInfo(){**

**System.out.println("Patient Id: " + ID) ;**

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

**System.out.println("Gender: " + gender) ;**

**System.out.println("Age: " + age) ;**

**System.out.println("Mobile Number: " + mobileNumber) ;**

**System.out.println("Address: " + address + "\n") ;**

**}**

**public boolean equals(Object obj){**

**if(this == obj)**

**return true ;**

**if(obj == null || getClass() != obj.getClass())**

**return false ;**

**Person other = (Person) obj ;**

**return this.ID.equals(other.ID) ;**

**}**

**}**

**package hospital.patients ;**

**import hospital.person.Person ;**

**public class PatientInfo extends Person{**

**public Person[] patients = new Person[10] ;**

**int nbPatients = 0 ;**

**public PatientInfo(String patientId,String name,String gender,int age,String mobileNumber,**

**String address){**

**super(patientId,name,gender,age,mobileNumber,address) ;**

**}**

**public PatientInfo(){} ;**

**public void displayInfo(){**

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

**System.out.println("\tPatient Information") ;**

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

**super.displayInfo() ;**

**}**

**public void addPatient(Person person){**

**if(person != null){**

**patients[nbPatients++] = person ;**

**}**

**}**

**public void removePatient(Person person){**

**int j = 0 ;**

**for(int i = 0 ; i < nbPatients ; i++){**

**if(person != null && !person.equals(patients[i])){**

**patients[j++] = patients[i] ;**

**}**

**}**

**nbPatients = j ;**

**}**

**}**

**package hospital.patients ;**

**import hospital.patients.PatientInfo ;**

**public class MedicalHistory{**

**private PatientInfo[] patients ;**

**public MedicalHistory(PatientInfo[] patients){**

**this.patients = patients ;**

**}**

**public void displayPastRecord(PatientInfo patient){**

**for(int i = 0 ; i < patients.length ; i++){**

**if(patients[i] != null && patient.equals(patients[i])){**

**patients[i].displayInfo() ;**

**}**

**}**

**}**

**}**

**package hospital.staff ;**

**import hospital.person.Person ;**

**public class Doctor extends Person{**

**protected String qualification ;**

**protected String specializaion ;**

**protected String consultation ;**

**Person[] doctors = new Person[10] ;**

**int nbDoctors = 0 ;**

**public Doctor(String ID,String name,String gender,int age,String mobileNumber,**

**String address,String qualification,String specializaion,String consultation){**

**super(ID,name,gender,age,mobileNumber,address) ;**

**this.qualification = qualification ;**

**this.specializaion = specializaion ;**

**this.consultation = consultation ;**

**}**

**public void displayInfo(){**

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

**System.out.println("\tDoctor Information") ;**

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

**super.displayInfo() ;**

**System.out.println("Qualification: " + qualification) ;**

**System.out.println("Specialization: " + specializaion) ;**

**System.out.println("Consultation time: " + consultation) ;**

**}**

**public void addDoctor(Person person){**

**if(person != null){**

**doctors[nbDoctors++] = person ;**

**}**

**}**

**public void removeDoctor(Person person){**

**int j = 0 ;**

**for(int i = 0 ; i < nbDoctors ; i++){**

**if(person != null && !person.equals(doctors[i])){**

**doctors[j++] = doctors[i] ;**

**}**

**}**

**nbDoctors = j ;**

**}**

**}**

**package hospital.staff ;**

**import hospital.person.Person ;**

**public class Nurse extends Person{**

**String qualification ;**

**String specialization ;**

**int experience ;**

**Person[] nurses = new Person[10] ;**

**int nbNurses = 0 ;**

**public Nurse(String ID,String name,String gender,int age,String mobileNumber,**

**String address,String qualification,String specialization,int experience){**

**super(ID,name,gender,age,mobileNumber,address) ;**

**this.qualification = qualification ;**

**this.specialization = specialization ;**

**this.experience = experience ;**

**}**

**public void displayInfo(){**

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

**System.out.println("\tNurse Information") ;**

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

**super.displayInfo() ;**

**System.out.println("Qualification: " + qualification) ;**

**System.out.println("Specialization: " + specialization) ;**

**System.out.println("Experience " + experience) ;**

**}**

**public void addNurse(Person person){**

**if(person != null){**

**nurses[nbNurses++] = person ;**

**}**

**}**

**public void removeDoctor(Person person){**

**int j = 0 ;**

**for(int i = 0 ; i < nbNurses ; i++){**

**if(person != null && !person.equals(nurses[i])){**

**nurses[j++] = nurses[i] ;**

**}**

**}**

**nbNurses = j ;**

**}**

**}**

**package hospital.person ;**

**import hospital.person.Person ;**

**public class HospitalUtils extends Person{**

**private String doctID ;**

**private double Bill ;**

**public HospitalUtils(String ID,String name,String gender,int age,String mobileNumber,**

**String address,String doctID,double Bill){**

**super(ID,name,gender,age,mobileNumber,address) ;**

**this.doctID = doctID ;**

**this.Bill = Bill ;**

**}**

**public void setDoctID(String doctID){**

**this.doctID = doctID ;**

**}**

**public void setBill(double Bill){**

**this.Bill = Bill ;**

**}**

**public String getDoctID(){**

**return doctID ;**

**}**

**public double getBill(){**

**return Bill ;**

**}**

**}**

**import hospital.person.Person;**

**import hospital.person.HospitalUtils;**

**import hospital.patients.PatientInfo;**

**import hospital.patients.MedicalHistory;**

**import hospital.staff.Doctor;**

**import hospital.staff.Nurse;**

**public class PackagePractice {**

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

**PatientInfo patient = new PatientInfo("P001", "Shakir", "Male", 20 ,**

**"0000000000", "Neosheroeroze");**

**Doctor doctor = new Doctor("D001", "Dr. Shahzaib", "Male", 20 , "0000000000",**

**"Neosheroeroze", "MBBS", "Cardiologist", "9AM-12PM");**

**Nurse nurse = new Nurse("N001", "Sahil", "Male", 20, "00000000000",**

**"Dado", "BSc Nursing", "General Care", 10);**

**HospitalUtils visit = new HospitalUtils("P001", "Shakir", "Male", 20 , "0000000000",**

**"Neosheroeroze", "D001", 2500.00);**

**patient.addPatient(patient);**

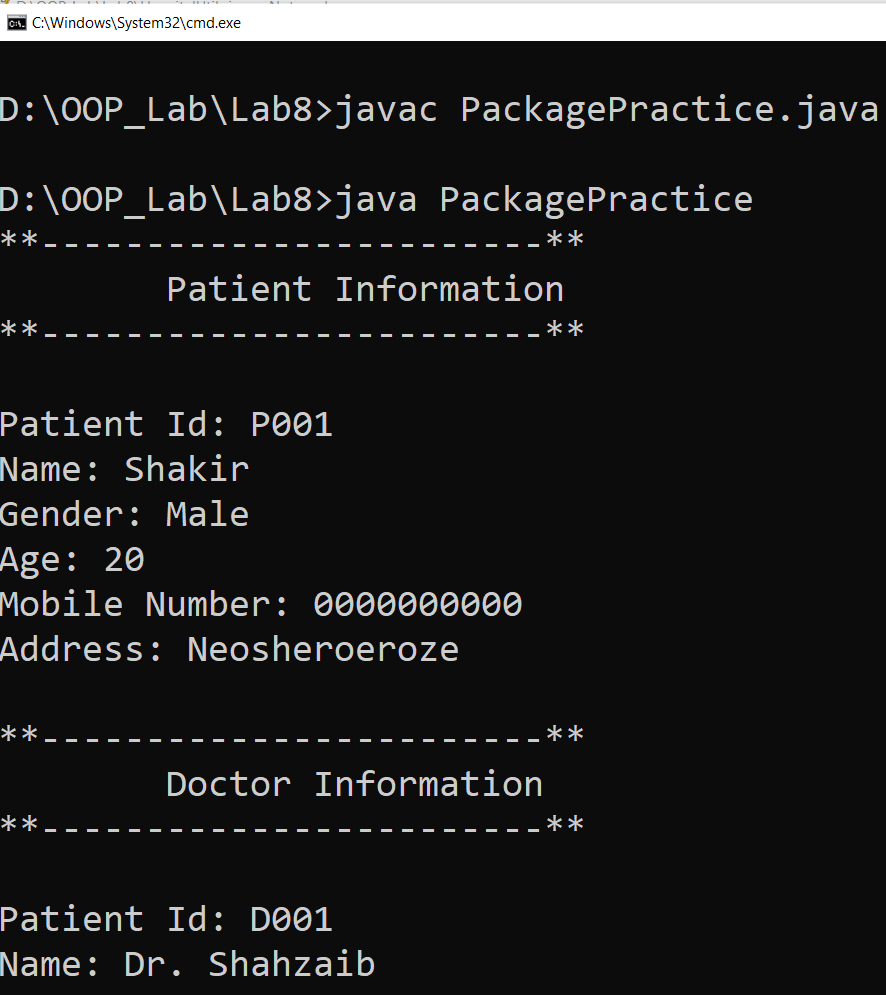
**patient.displayInfo();**

**doctor.displayInfo();**

**nurse.displayInfo();**

**MedicalHistory history = new MedicalHistory(new PatientInfo[]{patient});**

**System.out.println("\n--- Displaying Medical History ---");**

** history.displayPastRecord(patient);**

**System.out.println("\n--- Billing & Appointment Details ---");**

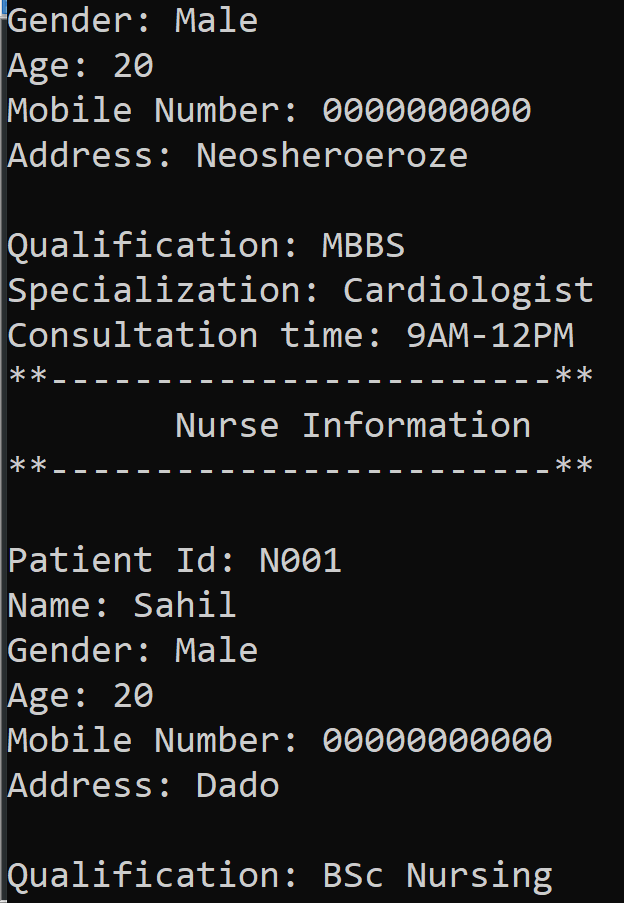
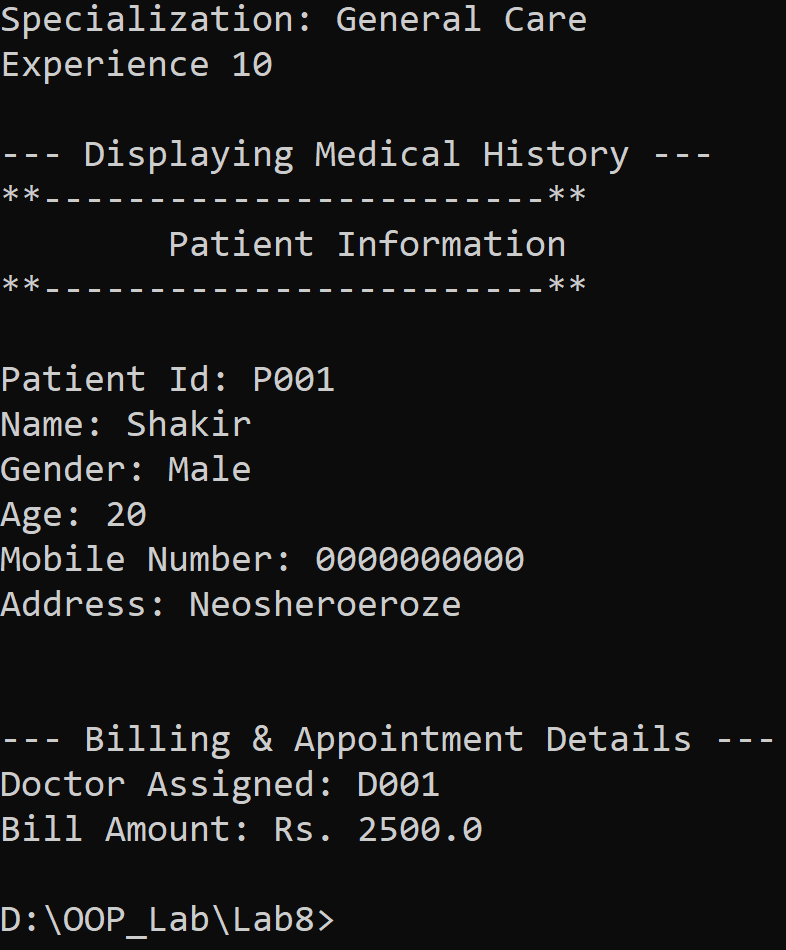
**System.out.println("Doctor Assigned: " + visit.getDoctID());**

**System.out.println("Bill Amount: Rs. " + visit.getBill());**

**}**

**}**

**Output:**

****

**The End**