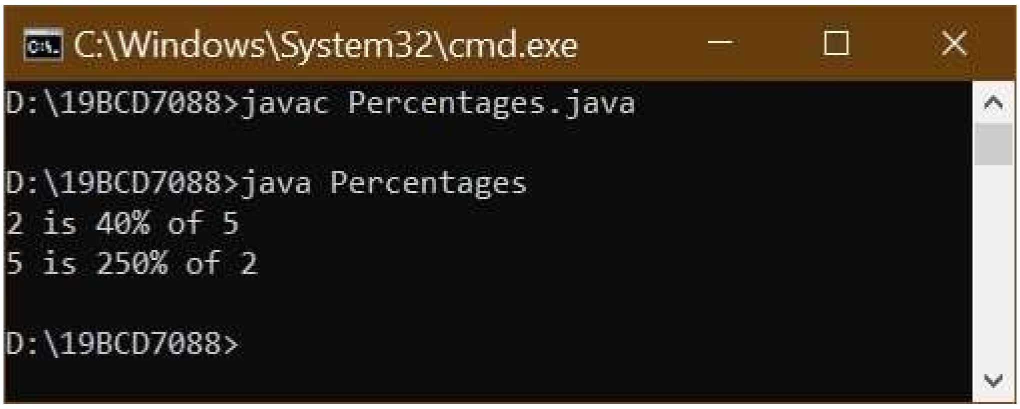
1.a.

Create an application named Percentages whose main() method holds two double variables. Assign values to the variables. Pass both variables to a method named computePercent() that displays the two values and the value of the first number as a percentage of the second one. For example, if the numbers are 2.0 and 5.0, the method should display a statement similar to “2.0 is 40 percent of 5.0.” Then call the method a second time, passing thevalues in reverse order. Save the application as Percentages.java.

```
public class Percentages{

            static void computePercent(int a, int b){

            System.out.println(a+" is "+(a*100/b)+"% of "+b);

      }

      public static void main(String[] args) {

            int a = 2;

            int b = 5;

            computePercent(2,5);

            computePercent(5,2);

      }


}
```

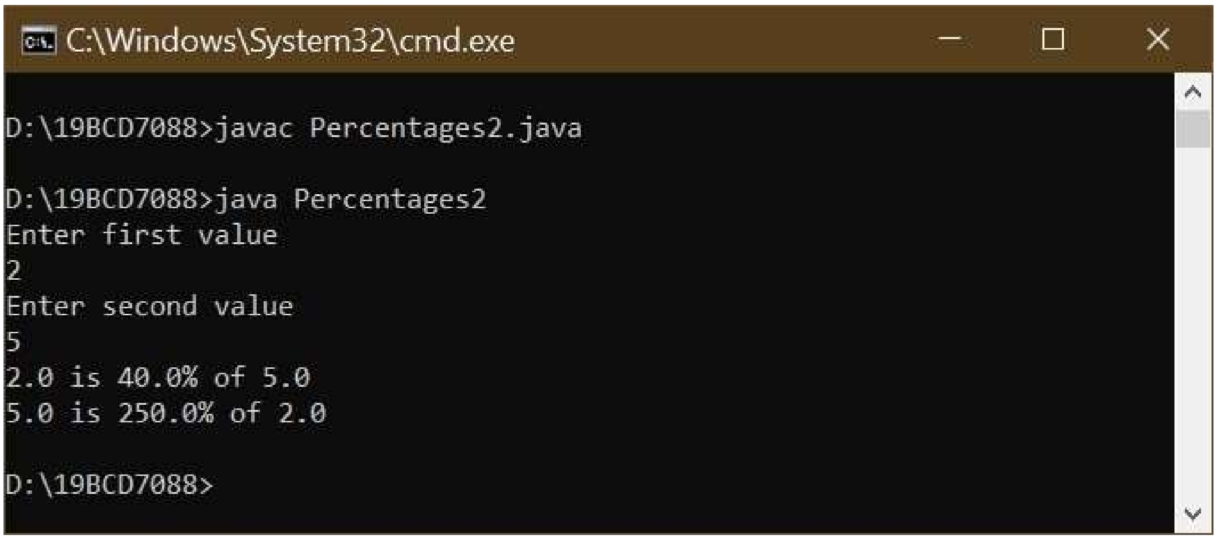


```
D:\19BCD7088>javac Percentages.java

D:\19BCD7088>java Percentages
2 is 40% of 5
5 is 250% of 2

D:\19BCD7088>
```

1.b.

Modify the Percentages class to accept the values of the two doubles from auser at the keyboard. Save the file as Percentages2.java.

```
import java.util.*;


public class Percentages2{
	static void computePercent(double a, double b){
		System.out.println(a+" is "+(a*100/b)+"% of "+b);
	}
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter first value");
		double a = sc.nextDouble();
		System.out.println("Enter second value");
		double b = sc.nextDouble();
		computePercent(a,b);
		computePercent(b,a);
	}

}
```

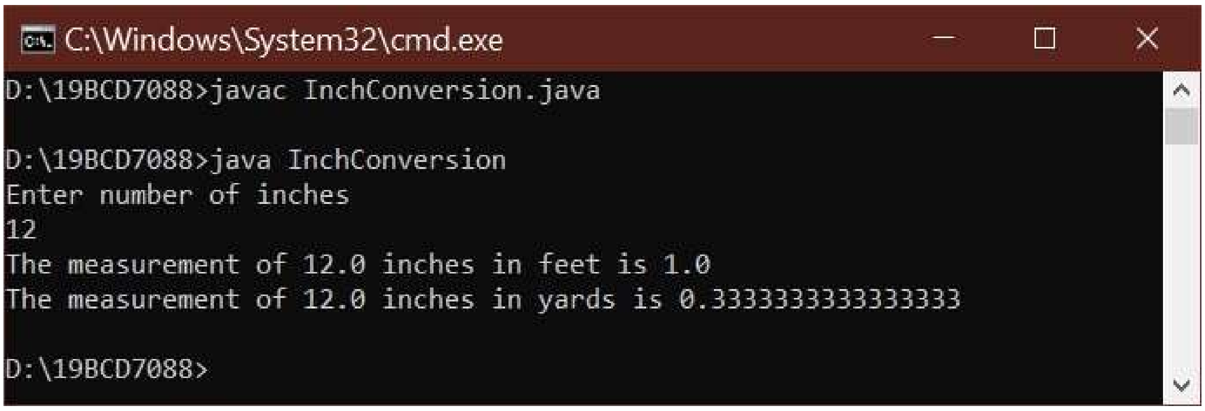


```
D:\19BCD7088>javac Percentages2.java

D:\19BCD7088>java Percentages2
Enter first value
2
Enter second value
5
2.0 is 40.0% of 5.0
5.0 is 250.0% of 2.0

D:\19BCD7088>
```

2.

There are 12 inches in a foot and 3 feet in a yard. Create a class named InchConversion. Its main() method accepts a value in inches from a user at the keyboard, and in turn passes the entered value to two methods. One converts the value from inches to feet, and the other converts the same value from inches to yards. Each method displays the results with appropriate explanation. Save the application as InchConversion.java.

```
import java.util.*;
public class InchConversion{
        static void InchtoFeet(double a){
                System.out.println("The measurement of "+ a +" inches in feet is "+ a/12);
        }
        static void InchtoYards(double b){
                System.out.println("The measurement of "+ b +" inches in yards is "+ b/36);
        }
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter number of inches");
                double a = sc.nextDouble();
                InchtoFeet(a);
                InchtoYards(a);
        }
}
```



```
D:\19BCD7088>javac InchConversion.java

D:\19BCD7088>java InchConversion
Enter number of inches
12
The measurement of 12.0 inches in feet is 1.0
The measurement of 12.0 inches in yards is 0.3333333333333333

D:\19BCD7088>
```

3.

Assume that a gallon of paint covers about 350 square feet of wall space. Create an application with a main() method that prompts the user for the length, width, and height of a rectangular room. Pass these three values to a method that does the following:

•• Calculates the wall area for a room

•• Passes the calculated wall area to another method that calculates and returns the number of gallons of paint needed

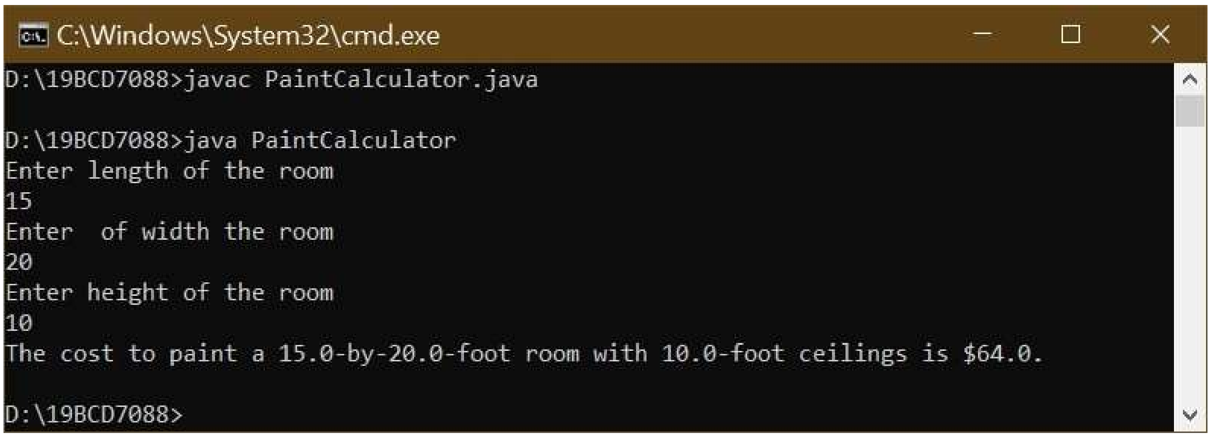
•• Displays the number of gallons needed

•• Computes the price based on a paint price of $32 per gallon, assuming that the painter can buy any fraction of a gallon of paint at the same price as a whole gallon

•• Returns the price to the main() method

The main() method displays the final price. For example, the cost to paint a 15-by-20-foot room with 10-foot ceilings is $64. Save the application as PaintCalculator.java.

```
import java.util.*;

public class PaintCalculator{
	static double area (double l,double b,double h){
		return ((2*l*h)+(2*b*h));
	}
	static double gallons(double q){
		return q/350;
	}
	static double price(double g){
		return g*32;
	}
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter length of the room");
		double l = sc.nextDouble();
		System.out.println("Enter of width the room");
		double b = sc.nextDouble();
		System.out.println("Enter height of the room");
		double h = sc.nextDouble();
		double a = area(l,b,h);
		double gallon = gallons(a);
		double rate = price(gallon);
		System.out.println("The cost to paint a " + l + "-by-" + b + "-foot room with " + h + "-
foot ceilings is $" + rate + ".");
	}
}
```



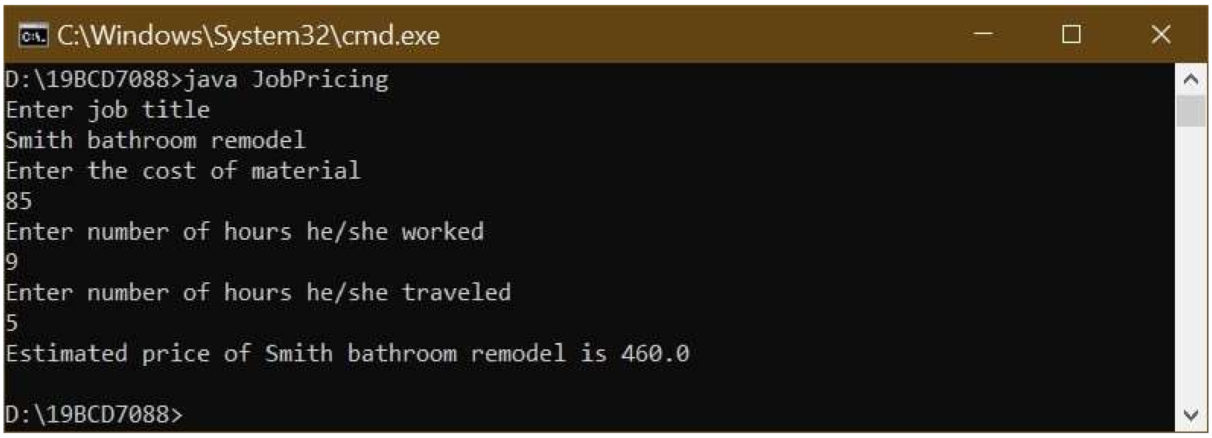
```
D:\19BCD7088>javac PaintCalculator.java

D:\19BCD7088>java PaintCalculator
Enter length of the room
15
Enter  of width the room
20
Enter height of the room
10
The cost to paint a 15.0-by-20.0-foot room with 10.0-foot ceilings is $64.0.

D:\19BCD7088>
```

4.

Herbert’s Home Repair estimates each job cost as the cost of materials plus $35 per hour while on the job, plus $12 per hour for travel time to the job site. Create a class that contains a main() method that prompts the user for the name of a job (for example, Smith bathroom remodel), the cost of materials, the number of hours of work required, and the number of hours travel time. Pass the numeric data to a method that computes estimate for the job and returns the computed value to the main() method where the job name and estimated price are

displayed. Save the program as JobPricing.java.

```
        import java.util.*;
        public class JobPricing{
                static double computetion (double m,double h,double t){
                        return m+(h*35)+(t*12);
                }
                public static void main(String[] args) {
                        Scanner sc = new Scanner(System.in);

                        System.out.println("Enter job title");

                        String str = sc.nextLine();

                        System.out.println("Enter the cost of material");

                        double m  = sc.nextDouble();

                        System.out.println("Enter number of hours he/she worked");

                        double h  = sc.nextDouble();

                        System.out.println("Enter number of hours he/she traveled");

                        double t  = sc.nextDouble();

                        double n = computetion(m,h,t);

                        System.out.println("Estimated price of " + str + " is " + n);
                }
}
```



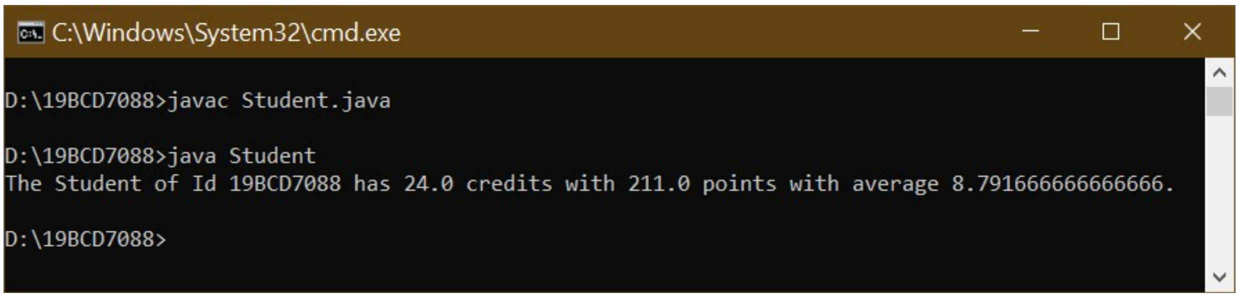
```
D:\19BCD7088>java JobPricing
Enter job title
Smith bathroom remodel
Enter the cost of material
85
Enter number of hours he/she worked
9
Enter number of hours he/she traveled
5
Estimated price of Smith bathroom remodel is 460.0

D:\19BCD7088>
```

5.a.

Create a class named Student that has fields for an ID number, number of credit hours earned, and number of points earned. (For example, many schools compute grade point averages based on a scale of 4, so a three-credit-hour class in which a student earns an A is worth 12 points.) Include methods to assign values to all fields. A Student also has a field for grade point average. Include a method to compute the grade point average field by dividing points by credit hours earned. Write methods to display the values in each Student field. Save this class as Student.java.

```
public class Student{
        static String str;
        static double
         credit,points; static void
         setId(String s){
                str=s;
        }
        static void setCredit(int c){
                credit=c;
        }
        static void setPoints(int p){
                points=p;
```

```
	}
	static void average(){
		double a = points/credit;
		System.out.println("The Student of Id " + str + " has " + credit + " credits with " +
points + " points with average " + a + ".");
	}
	public static void main(String[] args) {
		setId("19BCD7088");
		setCredit(24);
		setPoints(211);
		average();
	}
}
```



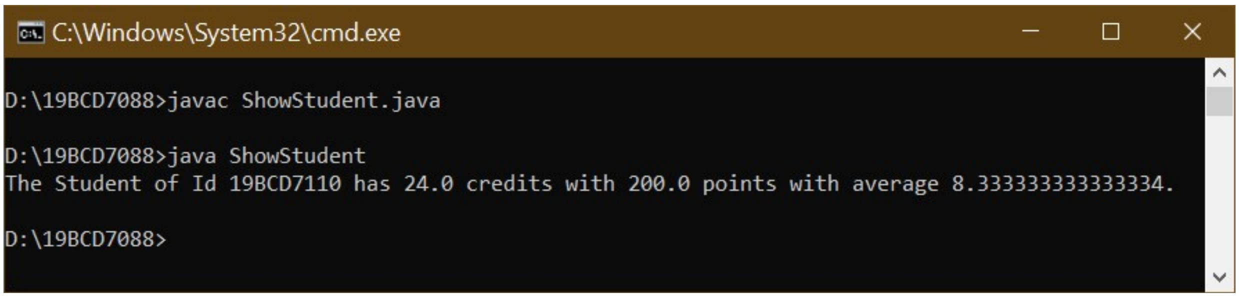
```
D:\19BCD7088>javac Student.java

D:\19BCD7088>java Student
The Student of Id 19BCD7088 has 24.0 credits with 211.0 points with average 8.791666666666666.

D:\19BCD7088>
```

5.b.

Write a class named ShowStudent that instantiates a Student object from the class you created and assign values to its fields. Compute the Student grade point average, and then display all the values associated with the Student. Save the application as ShowStudent.java.

```
class Student{
	String str;
	double
	  credit,points;

	 void setId(String s){
		str=s;
	}
```

```
        void setCredit(int c){
                credit=c;
        }
        void setPoints(int p){
                points=p;
        }
        void average(){
                double a = points/credit;
                System.out.println("The Student of Id " + str + " has " + credit + " credits with " +
points + " points with average " + a + ".");
        }
}

public class ShowStudent{
        public static void main(String[] args) {
                Student st = new Student();
                st.setId("19BCD7110");
                st.setCredit(24);
                st.setPoints(200);
                st.average();
        }
}
```

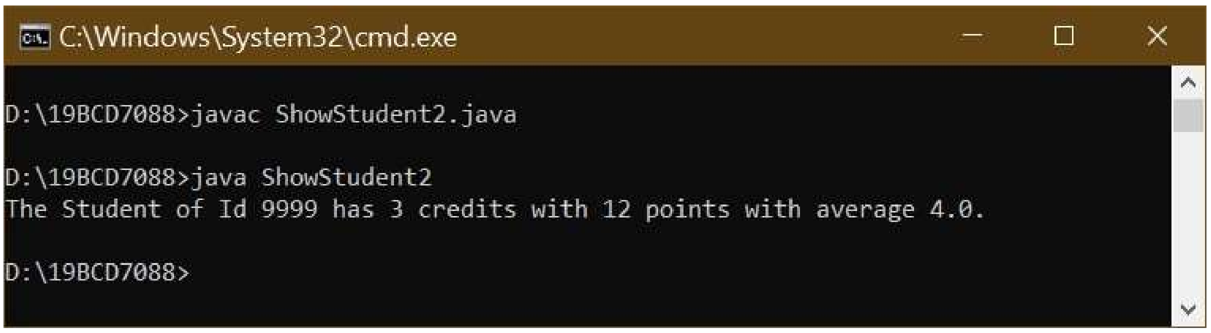


```
D:\19BCD7088>javac ShowStudent.java

D:\19BCD7088>java ShowStudent
The Student of Id 19BCD7110 has 24.0 credits with 200.0 points with average 8.333333333333334.

D:\19BCD7088>
```

5.c.

Create a constructor for the Student class you created. The constructor should initialize each Student’s ID number to 9999, his or her points earned to 12, and credit hours to 3 (resulting in a grade point average of 4.0). Write a program that demonstrates that the constructor works by instantiating an object and displaying the initial values. Save the application as ShowStudent2.java.

```
class Student{
        String str;
        double credit,points;
        Student(String s,int c,int p){
                str=s;
                credit=c;
                points=p;
        }
        void mean(){
                double a = points/credit;
                System.out.println("The Student of Id " + str + " has " + credit + " credits with " +
points + " points with average " + a + ".");
        }
}
public class ShowStudent2{
        public static void main(String[] args) {
        Student st = new Student("9999",3,12);
        st.mean();
}


}
```

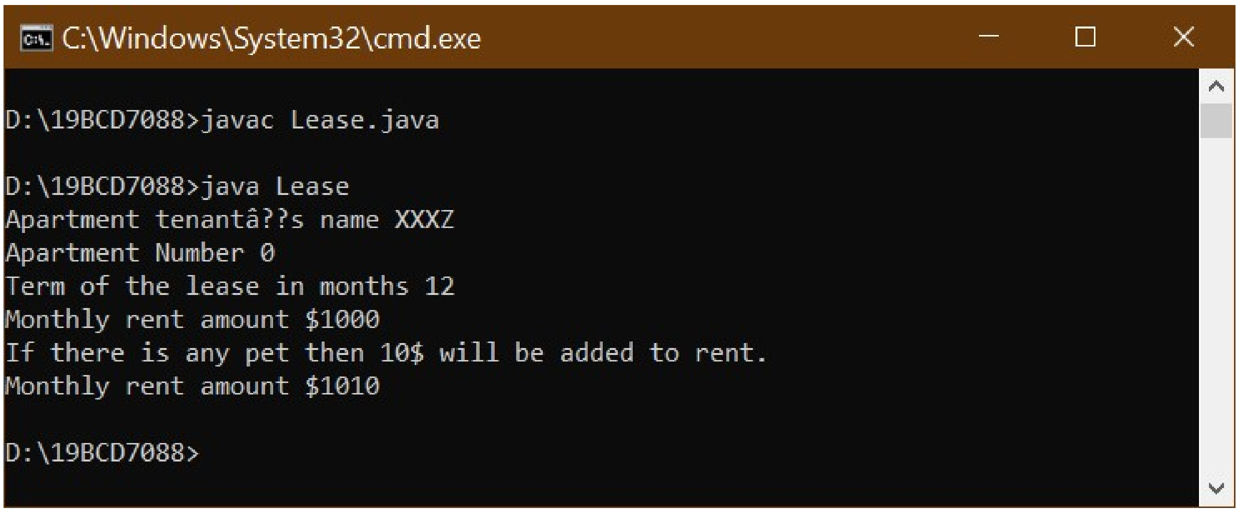


```
D:\19BCD7088>javac ShowStudent2.java

D:\19BCD7088>java ShowStudent2
The Student of Id 9999 has 3 credits with 12 points with average 4.0.

D:\19BCD7088>
```

6.a.

Create a class named Lease with fields that hold an apartment tenant’s name, apartment number, monthly rent amount, and term of the lease in months. Include a constructor that initializes the name to “XXX”, the apartment number to 0, the rent to 1000, and the term to 12. Also include methods to get and set each of the fields. Include a nonstatic method named addPetFee() that adds $10 to the monthly rent value and calls a static method named explainPetPolicy() that explains the pet fee. Save the class as Lease.java.

```
class Data{

        String str;

        int a,r,m;

        Data(String str,int a ,int r, int m){

                this.str=str;

                this.a=a;

                this.r=r;

                this.m=m;

        }

        void setName(String s){

                this.str=s;

        }

        void setAptNum(int a){

                this.a=a;

        }

        void setRent(int r){

                this.r=r;

        }

        void setMonth(int m){

                this.m=m;

        }

        void getName(){

                System.out.println("Apartment tenant’s name " + this.str);

        }

        void getAptNum(){
                System.out.println("Apartment Number " + this.a);
        }
        void getMonth(){
                System.out.println("Term of the lease in months " + this.m);
        }
        void getRent(){
                System.out.println("Monthly rent amount $" + this.r);
        }
        void addPetFee(){
                this.r= this.r+10;
        }
        static void explainPetPolicy(){
                System.out.println("If there is any pet then 10$ will be added to rent.");
        }

}
public class Lease{
        public static void main(String[] args) {
                Data d = new Data("XXXZ",0,1000,12);
                d.getName();
                d.getAptNum();
                d.getMonth();
                d.getRent();
                d.explainPetPolicy();
                d.addPetFee();
                d.getRent();



        }
}
```



```
D:\19BCD7088>javac Lease.java

D:\19BCD7088>java Lease
Apartment tenantâ??s name XXXZ
Apartment Number 0
Term of the lease in months 12
Monthly rent amount $1000
If there is any pet then 10$ will be added to rent.
Monthly rent amount $1010

D:\19BCD7088>
```

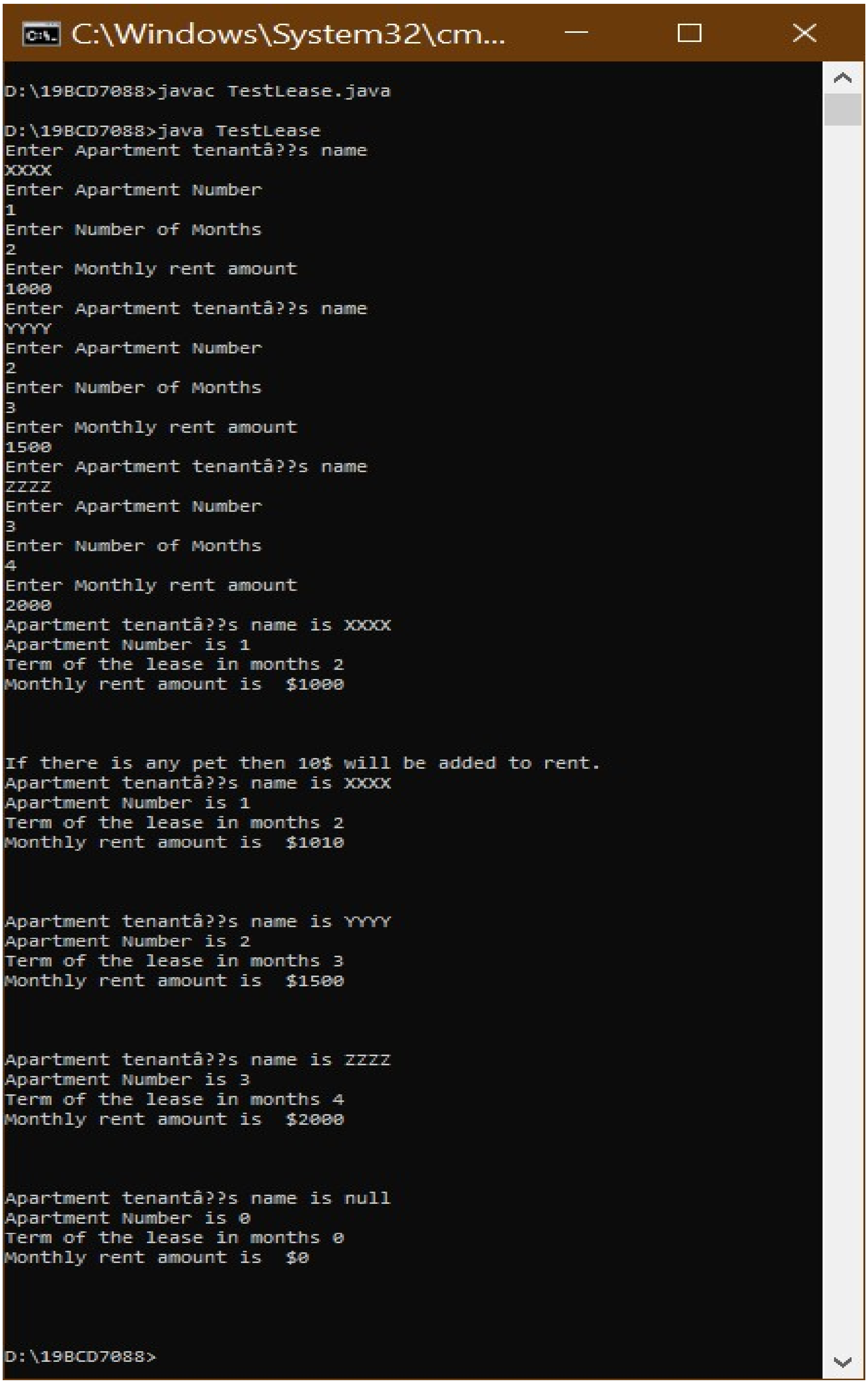
6.b.

Create a class named TestLease whose main() method declares four Lease objects. Call a getData() method three times. Within the method, prompt a user for values for each field for a Lease, and return a Lease object to the main() method where it is assigned to one of main()'s Lease objects. Do not prompt the user for values for the fourth Lease object, but let it continue to hold the default values. Then, in main(), pass one of the Lease objects to a showValues() method that displays the data. Then call the addPetFee() method using the passed Lease object and confirm that the fee explanation statement is displayed. Next, call the showValues() method for the Lease object again and confirm that the pet fee has been added to the rent. Finally, call the showValues() method with each of the other three objects; confirm that two hold the values you supplied as input and one holds the constructor default values. Save the application as TestLease.java.

```
import java.util.*;
class Lease{
        String str;
        int a,r,m;
        void setName(String s){
                this.str=s;
        }
        void setAptNum(int a){
                this.a=a;
        }
        void setRent(int r){
                this.r=r;
        }
        void setMonth(int m){
```

```
		this.m=m;

	}

	void getName(){

		System.out.println("Apartment tenant’s name is " + this.str);

	}

	void getAptNum(){

		System.out.println("Apartment Number is " + this.a);

	}

	void getMonth(){

		System.out.println("Term of the lease in months " + this.m);

	}

	void getRent(){

		System.out.println("Monthly rent amount is $" + this.r);

	}

	void addPetFee(){

		this.r= this.r+10;

	}

	Lease getData(){

		Scanner sc = new Scanner(System.in);

		Lease temp = new Lease();

		System.out.println("Enter Apartment tenant’s name");

		String e = sc.nextLine();

		System.out.println("Enter Apartment Number");

		int k =sc.nextInt();

		System.out.println("Enter Number of Months");

		int o = sc.nextInt();

		System.out.println("Enter Monthly rent amount");

		int l = sc.nextInt();

		temp.str=e;

		temp.a=k;

		temp.r=l;
```

```
                temp.m=o;
                return temp;


        }
        void showValues(){
                getName();
                getAptNum();
                getMonth();
                getRent();
                System.out.println("\n\n");
        }
        static void explainPetPolicy(){
                System.out.println("If there is any pet then 10$ will be added to rent.");
        }
}
public class TestLease{
        public static void main(String[] args) {
                Lease l1 = new Lease();
                Lease l2 = new Lease();
                Lease l3 = new Lease();
                Lease l4 = new Lease();
                l1=l1.getData();
                l2=l2.getData();
                l3=l3.getData();
                l1.showValues();
                l1.addPetFee();
                l1.explainPetPolicy();
                l1.showValues();
                l2.showValues();
                l3.showValues();
                l4.showValues();
        }
    }
```



C:\Windows\System32\cm...

```
D:\19BCD7088>javac TestLease.java

D:\19BCD7088>java TestLease
Enter Apartment tenantâ??s name
XXXX
Enter Apartment Number
1
Enter Number of Months
2
Enter Monthly rent amount
1000
Enter Apartment tenantâ??s name
YYYY
Enter Apartment Number
2
Enter Number of Months
3
Enter Monthly rent amount
1500
Enter Apartment tenantâ??s name
ZZZZ
Enter Apartment Number
3
Enter Number of Months
4
Enter Monthly rent amount
2000
Apartment tenantâ??s name is XXXX
Apartment Number is 1
Term of the lease in months 2
Monthly rent amount is  $1000



If there is any pet then 10$ will be added to rent.
Apartment tenantâ??s name is XXXX
Apartment Number is 1
Term of the lease in months 2
Monthly rent amount is  $1010



Apartment tenantâ??s name is YYYY
Apartment Number is 2
Term of the lease in months 3
Monthly rent amount is  $1500



Apartment tenantâ??s name is ZZZZ
Apartment Number is 3
Term of the lease in months 4
Monthly rent amount is  $2000



Apartment tenantâ??s name is null
Apartment Number is 0
Term of the lease in months 0
Monthly rent amount is  $0




D:\19BCD7088>
```