

Q.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 the values in reverse order. Save the application as Percentages.java.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
public class Percentages {
    public static void main(String[] args) {
        double num1 = 2.0;
        double num2 = 5.0;
        computePercent(num1,num2);
    }
    public static void computePercent(double num1,double num2)
    {
        double result1 = (num1/num2)*100;
        double result2 = (num2/num1)*100;
        System.out.println(num1+" is "+result1+ " percent of "
            +num2);
        System.out.println(num2+" is "+result2+ " percent of "
            +num1);
    }
}
```

```
java -cp /tmp/x22TlqG03F Percentages Output -
2.0 is 40.0 percent of 5.0
5.0 is 250.0 percent of 2.0
```

b) Modify the Percentages class to accept the values of the two doubles from a user at the keyboard. Save the file as Percentages2.java.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
import java.util.Scanner;
public class Percentages2 {
    public static void main(String[] args) {
        double num1;
        double num2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter number 1: ");
        num1 = sc.nextFloat();
        System.out.println("Enter number 2: ");
        num2 = sc.nextFloat();
        computePercent(num1,num2);
    }
    public static void computePercent(double num1,double num2)
    {
        double result1 = (num1/num2)*100;
        double result2 = (num2/num1)*100;
        System.out.println(num1+" is "+result1+ " percent of "
            +num2);
        System.out.println(num2+" is "+result2+ " percent of "
            +num1);
    }
}
```

Output -

```
java -cp /tmp/x22TlqG03F Percentages2
Enter number 1:
12
Enter number 2:
36
12.0 is 33.33333333333333 percent of 36.0
36.0 is 300.0 percent of 12.0
```

Q.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.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
import java.util.Scanner;
public class InchConversion {
    public static void main(String[] args) {

        double inches;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter value in inches: ");
        inches = sc.nextDouble();
        convertToFeet(inches);
        convertToYards(inches);
    }
    public static void convertToFeet(double inches) {
        int foot = 12;
        double result = inches / foot;
        System.out.println(+inches+" inches = "+result+ "
            feet");
    }
    public static void convertToYards(double inches) {
        double yard = 0.0278;
        double result = inches*yard;
        System.out.println(inches+ " inches = "+result+ "
            yards");
    }
}
```

Output –

```
java -cp /tmp/x22TlqG03F InchConversion
Enter value in inches: 12
12.0 inches = 1.0 feet
12.0 inches = 0.3336 yards
```

Q.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

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
import java.util.Scanner;
public class PaintCalculator {
    public static void main(String [] args)
    {
        Scanner keyboard = new Scanner(System.in);
        double wallArea;
        double height;
        double length;
        double width;
        double price;
        double WallArea;
        double paintQuantity;
        System.out.print("Please enter the height of the room: ");
        height = keyboard.nextDouble();
        System.out.print("Please enter the length of the room: ");
        length = keyboard.nextDouble();
        System.out.print("Please enter the width of the room: ");
        width = keyboard.nextDouble();
        WallAreaMethod(height, length, width);
    }
    public static double WallAreaMethod(double height, double length,
        double width)
    {
        double wallArea;
        wallArea = length * height * width * height;
        return wallArea;
    }
    public static double paintFormula(double wallAreaMethod, double
        price, double height, double length, double width)
    {
        double wallArea;
```

```

double paintQuantity;
paintQuantity = wallAreaMethod * 2 / 350;

System.out.println("For a room of height " + height + "feet, length
" + length + " feet, and width " + width + " feet you need to
purchase " + paintQuantity + " gallons of paint.");
System.out.println("The price will be $" + price + ".");
price = paintQuantity * 32.0;
return price;
}
}

```

Output –

```

java -cp /tmp/x22TlqG03F PaintCalculator
Please enter the height of the room: 15
Please enter the length of the room: 20
Please enter the width of the room: 25

```

Q.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.

```

/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
import java.util.Scanner;
public class JobPricing {
    public static void main(String[] args) {
        String description;
        double materials;
        double hoursOnJob;
        double hoursTraveling;
        double price;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter job description: ");
        description = sc.nextLine();
        System.out.print("Enter cost of materials: ");
        materials = sc.nextDouble();
        System.out.print("Enter hours on the job work: ");
        hoursOnJob = sc.nextDouble();
        System.out.print("Enter hours traveling: ");
    }
}

```

```

hoursTraveling = sc.nextDouble();
price = computePrice(materials, hoursOnJob, hoursTraveling);
System.out.println("The price for " + description + " is $" +
price);
}
public static double computePrice(double materials, double
hoursOnJob, double hoursTraveling)
{
double price = materials + (hoursOnJob *35) + (hoursTraveling * 12);
return price;
}
}

```

Output –

```

java -cp /tmp/xVPqu1X88p JobPricing
Enter job description: 12
Enter cost of materials: 25
Enter hours on the job work: 8
Enter hours traveling: 1
The price for 12 is $317.0

```

Q.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.

```

/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
public class Student {
    int idNumber;
    int hours;
    int points;
    public Student(){
        idNumber = 7138;
        hours = 3;
    }
}

```

```
        points = 12;
    }
    public int getIdNumber(){
        return idNumber;
    }
    public void setIdNumber(int num_ID){
        idNumber = num_ID;
    }
    public int getHours(){
        return hours;
    }
    public void setHours(int num_hours){
        hours = num_hours;
    }
    public int getPoints(){
        return points;
    }
    public void setPoints(int num_point){
        points = num_point;
    }
    public void showIdNumber() {
        System.out.println("Student ID: "+getIdNumber());
    }
    public void showHours() {
        System.out.println("Hours: "+getHours());
    }
    public void showPoints() {
        System.out.println("Points: "+getPoints());
    }
    public double getGradePoint() {
```

```
        double result = getPoints()/getHours();
        return result;
    }
}
```

Class : –

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
class Student
{
    private int IDnumber;
    private int hours;
    private int points;
    Student()
    {
        IDnumber = 7138;
        points = 12;
        hours = 3;
    }
    public void setIDnumber(int number)
    {
        IDnumber = number;
    }
    public int getPoints()
    {
        return points;
    }
    public void showIDnumber()
    {
        System.out.println("ID Number is: " + IDnumber);
    }
}
```



```

    }

    public void showHours()
    {
        System.out.println("Credit Hours: " + hours);
    }

    public void showPoints()
    {
        System.out.println("Points Earned: " + points);
    }

    public double getGradePoint()
    {
        return (double)points / hours;
    }
}

```

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.

```

/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
public class ShowStudent {

    public static void main(String[] args) {
        ShowStudent stud = new ShowStudent();
        stud.setIdNumber(7138);
        stud.setHours(3);
        stud.setPoints(12);

        stud.showIdNumber();
        stud.showHours();
        stud.showPoints();
        System.out.println("The grade point average is " +
            stud.getGradePoint());
    }
}

```

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.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
public class ShowStduent2 {
    public static void main(String[] args) {
        ShowStudent2 stud = new ShowStudent2();
        stud.showIdNumber();
        stud.showHours();
        stud.showPoints();

        System.out.println("The grade point average is " + stud
            .getGradePoint());
    }
}
```

Q.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.

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
public class Lease {
    private String name;
    private int aptNumber;
    private double rent;
    private int term;
    private static final int FEE = 10;
    public Lease() {
        name = "XXX";
        aptNumber = 0;
```

```
        rent = 1000;
        term = 12;
    }
    public void setName(String tenant) {
        this.name = tenant;
    }
    public void setAptNumber(int apt) {
        this.aptnumber = apt;
    }
    public void setRent(double monthRent) {
        this.rent = monthRent;
    }
    public void setTerm(int t) {
        this.term = t;
    }
    public String getName() {
        return name;
    }
    public int getAptNumber() {
        return aptNumber;
    }

    public double getRent() {
        return rent;
    }
    public int getTerm() {
        return term;
    }
    public void addPetFee() {
        explainPetPolicy();
    }
}
```

```

        double new_rent = +FEE;

        System.out.println("New Rent: "+new_rent);
    }

    public static void explainPetPolicy() {
        System.out.println("A pet fee of $10 is added to the monthly
rent");
    }
}

```

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 .

```

/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 Lab-1 */
import java.util.Scanner;
public class TestLease {
    public static void main (String args[])
    {
        Lease one = new Lease();
        Lease two = new Lease();
        Lease three = new Lease();
        Lease four = new Lease();
        one = getData();
        two = getData();
        three = getData();
        showValues(one);
        one.addPetFee();
        showValues(one);
        showValues(two);
        showValues(three);
        showValues(four);
    }
}

```

```
public static void showValues(Lease ls)
{
    System.out.println("\n\nYour lease results:" +
        "\nName      : " + ls.getName() +
        "\nApartment : " + ls.getAptNumber() +
        "\nRent      : " + ls.getRent() +
        "\nTerm      : " + ls.getTerm());
}
public static Lease getData() {
    Lease lse = new Lease();
    String name;
    int aptNumber;
    double rent;
    int term;
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter name of the tenant: ");
    name = sc.nextLine();
    lse.setName(name);
    System.out.println("Enter the apartment number: ");
    aptNumber = sc.nextInt();
    lse.setAptNumber(aptNumber);
    System.out.println("Enter the rent: ");
    rent = sc.nextDouble();
    lse.setRent(rent);
    System.out.println("Enter the lease term: ");
    term = sc.nextInt();
    lse.setTerm(term);
    return lse;
}
}
```