

Name : KHAN MOHD. OWAIS RAZA**Reg. No. : 20BCD7138**

Q.1] Create a class named Billing that includes three overloaded computeBill() methods for a photo book store.

- When computeBill() receives a single parameter, it represents the price of one photo book ordered. Add 8% tax, and return the total due.
- When computeBill() receives two parameters, they represent the price of a photo book and the quantity ordered. Multiply the two values, add 8% tax, and return the total due.
- When computeBill() receives three parameters, they represent the price of a photo book, the quantity ordered, and a coupon value. Multiply the quantity and price, reduce the result by the coupon value, and then add 8% tax and return the total due. Write a main() method that tests all three overloaded methods. Save the application as Billing.java.

Java program : –

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-2*/
public class Billing
{
    public static double computeBill(double Price)
    {
        double taxes = 0.08*Price;
        return Price+taxes;
    }
    public static double computeBill(double Price, int quant) {
        double taxes = 0.08*(Price*quant);
        return (quant*Price)+taxes;
    }
    public static double computeBill(double Price, int quant,double
    value) {
        double taxes = 0.08*(Price*quant);
        return ((quant*Price)+taxes)-value;
    }
    public static void main(String args[]) {
        System.out.println(computeBill(10));
        System.out.println(computeBill(10, 2));
        System.out.println(computeBill(10, 20, 50));
    }
}
```

Output : –

```
10.8
21.6
166.0
```

```
** Process exited - Return Code: 0 **
```

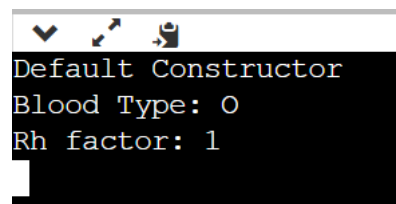
Q.2]

a) Create a class named BloodData that includes fields that hold a blood type (the four blood types are O, A, B, and AB) and an Rh factor (the factors are + and -). Create a default constructor that sets the fields to O and +, and an overloaded constructor that requires values for both fields. Include get and set methods for each field. Save this file as BloodData.java. Create an application named TestBloodData that demonstrates each method works correctly. Save the application as TestBloodData.java.

TestBloodData (java program) :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.util.*;
public class TestBloodData
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        BloodData bd=new BloodData();
        String bt=sc.nextLine();
        String rh=sc.nextLine();
        BloodData bd1=new BloodData(bt,rh);
        System.out.println("Setting data to AB and '-'");
        bd1.setdata("AB","-");
        System.out.println("Blood Type: "+bd1.getBt()+"\nRh factor: "+bd1.getRh());
    }
}
class BloodData
{
    String BloodType,Rhfact;
    BloodData()
    {
        System.out.println("Default Constructor");
        BloodType="O";
        Rhfact="1";
        System.out.println("Blood Type: "+BloodType+"\nRh factor: "+Rhfact);
    }
    BloodData(String Bt,String Rf)
    {
        System.out.println("Constructor");
        BloodType=Bt;
        Rhfact=Rf;
        System.out.println("Blood Type: "+BloodType+"\nRh factor: "+Rhfact);
    }
    void setdata(String Bt,String Rf)
    {
        BloodType=Bt;
        Rhfact=Rf;
    }
    String getBt()
    {
        return BloodType;
    }
    String getRh()
    {
        return Rhfact;
    }
}
```

Output :-



```
Default Constructor
Blood Type: O
Rh factor: 1
```

b) Create a class named Patient that includes an ID number, age, and BloodData. Provide a default constructor that sets the ID number to 0, the age to 0, and the BloodData values to O and +. Create an overloaded constructor that provides values for each field. Also provide get methods for each field. Save the file as Patient.java. Create an application that demonstrates that each method works correctly, and save it as TestPatient.java.

Patient class (java program) :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-2 */
public class Patient {
    private String id;
    private int age;
    private BloodData bloodData;
    public Patient() {
        id = "0";
        age = 0;
        setBloodData(new BloodData("O","+"));
    }
    public Patient(String id, int age, String bType, String rhFactor) {
        this.id = id;
        this.age = age;
        this.setBloodData(new BloodData(bType,rhFactor));
    }
    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public BloodData getBloodData() {
        return bloodData;
    }
    public void setBloodData(BloodData b) {
        this.bloodData = b;
    }
}
```

Q.4] Write a Java application that uses the Math class to determine the answers for each of the following:

- The square root of 37
- The sine and cosine of 300
- The value of the floor, ceiling, and round of 22.8
- The larger and the smaller of the character 'D' and the integer 71
- A random number between 0 and 20

(Hint: The random() method returns a value between 0 and 1; you want a number that is 20 times larger.) Save the application as MathTest.java.

Java Program :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.util.Random;
public class MathTest {
    public static void main(String[] args) {
        double sq = Math.sqrt(37);
        System.out.println("The square root of 37: "+sq);
        double sine = Math.sin(300);
        double cosine = Math.cos(300);
        System.out.println("The sine of 300: "+sine+" and cosine of 300: "+cosine);
        System.out.println("Floor value of 22.8: "+Math.floor(22.8));
        System.out.println("Celing value of 22.8: "+Math.ceil(22.8));
        System.out.println("Round value of 22.8: "+Math.round(22.8));
        System.out.println("The larger of D and 71 is " + Math.max('D', 71));
        System.out.println("The smaller of D and 71 is " + Math.min('D', 71));
        System.out.println("A random number between 0 and 20 is " + Math.random() * 20 + 1);
    }
}
```

Output :-

```
The square root of 37: 6.082762530298219
The sine of 300: -0.9997558399011495 and cosine of 300: -0.022096619278683942
Floor value of 22.8: 22.0
Celing value of 22.8: 23.0
Round value of 22.8: 23
The larger of D and 71 is 71
The smaller of D and 71 is 68
A random number between 0 and 20 is 7.7316332801821271

** Process exited - Return Code: 0 **
```

Q.5]

a) Write a program that declares two LocalDate objects and assign values that represent January 31 and December 31 in the current year. Display output that demonstrates the dates displayed when one, two, and three months are added to each of the objects. Save the application as TestMonthHandling.java.

TestMonthHandling (Java Program) :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.time.*;
public class TestMonthHandling {
    public static void main(String[] args) {
        LocalDate jan31 = LocalDate.of(2020,Month.JANUARY,31);
        LocalDate dec31 = LocalDate.of(2020,Month.DECEMBER,31);
        System.out.println(jan31);
        System.out.println("When one month is added: "+jan31.plusMonths(1)+"\t"+dec31.plusMonths(1));
        System.out.println("When two month is added: "+jan31.plusMonths(2)+"\t"+dec31.plusMonths(2));
        System.out.println("When three month is added: "+jan31.plusMonths(3)+"\t"+dec31.plusMonths(3));
        System.out.println(dec31);
    }
}
```

Output :-

```
2020-01-31
When one month is added: 2020-02-29      2021-01-31
When two month is added: 2020-03-31      2021-02-28
When three month is added: 2020-04-30    2021-03-31
2020-12-31

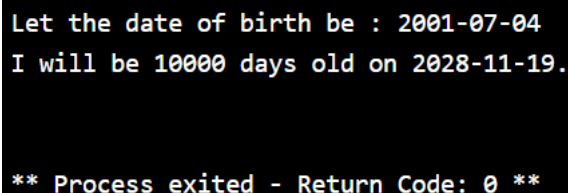
** Process exited - Return Code: 0 **
```

b) Write an application that computes and displays the day on which you become (or became) 10,000 days old. Save the application as TenThousandDaysOld.java.

TenThousandDaysOld (java program) :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.util.Scanner;
import java.time.*;
public class TenThousandDaysOld {
    public static void main(String[] args) {
        LocalDate a = LocalDate.of(2001,7,4);
        int DAYS = 10000;
        LocalDate futuredate = a.plusDays(DAYS);
        System.out.println("Let the date of birth be : "+a);
        System.out.println("I will be 10000 days old on "+futuredate+".");
    }
}
```

Output :-

A screenshot of a terminal window showing the output of the TenThousandDaysOld.java program. The output consists of two lines: "Let the date of birth be : 2001-07-04" and "I will be 10000 days old on 2028-11-19.". Below the output, it says "** Process exited - Return Code: 0 **".

```
Let the date of birth be : 2001-07-04
I will be 10000 days old on 2028-11-19.

** Process exited - Return Code: 0 **
```

c) The LocalDate class includes an instance method named lengthOfMonth() that returns the number of days in the month. Write an application that uses methods in the LocalDate class to calculate how many days are left until the first day of next month. Display the result, including the name of the next month. Save the file as DaysTilNextMonth.java.

DaysTilNextMonth (java program) :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
import java.time.*;
import java.util.Scanner;
public class DaysTilNextMonth
{
    public static void main(String[] args)
    {
        LocalDate today = LocalDate.now();
        int day;
        int daysTilNextMo;
        System.out.println("Today is " + today);
        day = today.getDayOfMonth();
        System.out.println(day);
        daysTilNextMo = today.lengthOfMonth() - day;
        System.out.println("There are "
            + daysTilNextMo +
            " days until "
            + today.plusMonths(1).getMonth() +
            " starts");
    }
}
```

Output :-

```
Today is 2021-11-30
30
There are 0 days until DECEMBER starts

** Process exited - Return Code: 0 **
```

Q.3]

a) Create a class named Circle with fields named radius, diameter, and area. Include a constructor that sets the radius to 1 and calculates the other two values. Also include methods named setRadius() and getRadius(). The setRadius() method not only sets the radius, but it also calculates the other two values. (The diameter of a circle is twice the radius, and the area of a circle is pi multiplied by the square of the radius. Use the Math class PI constant for this calculation.) Save the class as Circle.java.

b) Create a class named TestCircle whose main() method declares several Circle objects. Using the setRadius() method, assign one Circle a small radius value, and assign another a larger radius value. Do not assign a value to the radius of the third circle; instead, retain the value assigned at construction. Display all the values for all the Circle objects. Save the application as TestCircle.java.

```
Circle.java :-  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
                /* CSE2005 LAB-2 */
                class Circle {
                    private double radius;
                    private double area;
                    private double diameter;
                    Circle() {
                        radius = 1;
                        diameter = 2 * radius;
                        area = Math.PI * radius * radius;
                    }
                    public void setRadius(double r) {
                        this.radius = r;
                        diameter = 2 * radius;
                        area = Math.PI * radius * radius;
                    }
                    public double getRadius() {
                        return radius;
                    }
                    private void computeDiameter() {
                        diameter = 2*radius;
                    }
                    private void computeArea() {
                        area = Math.PI * radius * radius;
                    }
                    public double getDiameter() {
                        return diameter;
                    }
                    public double getArea() {
                        return area;
                    }
                }
```

TestCircle.java :-

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
public class TestCircle {
    public static void main(String[] args) {
        Circle c1 = new Circle();
        c1.setRadius(2);

        Circle c2 = new Circle();
        c2.setRadius(50);

        Circle c3 = new Circle();

        System.out.println("\nSMALL CIRCLE :");
        System.out.printf("Radius = %.2f\n", c1.getRadius());
        System.out.printf("Diameter = %.2f\n", c1.getDiameter());
        System.out.printf("Area = %.2f\n", c1.getArea());
        System.out.println("-----\n:");
        System.out.println("\nBIG CIRCLE : ");
        System.out.printf("Radius = %.2f\n", c2.getRadius());
        System.out.printf("Diameter = %.2f\n", c2.getDiameter());
        System.out.printf("Area = %.2f\n", c2.getArea());
        System.out.println("-----\n:");
        System.out.println("\nDEFAULT CIRCLE : ");
        System.out.printf("Radius = %.2f\n", c3.getRadius());
        System.out.printf("Diameter = %.2f\n", c3.getDiameter());
        System.out.printf("Area = %.2f\n", c3.getArea());
    }
}
```

Output :-

```
SMALL CIRCLE :
Radius = 2.00
Diameter = 4.00
Area = 12.57
BIG CIRCLE :
Radius = 50.00
Diameter = 100.00
Area = 7853.98
```



```

SMALL CIRCLE :
Radius = 2.00
Diameter = 4.00
Area = 12.57
-----
BIG CIRCLE :
Radius = 50.00
Diameter = 100.00
Area = 7853.98
-----
DEFAULT CIRCLE :
Radius = 1.00
Diameter = 2.00
Area = 3.14

```

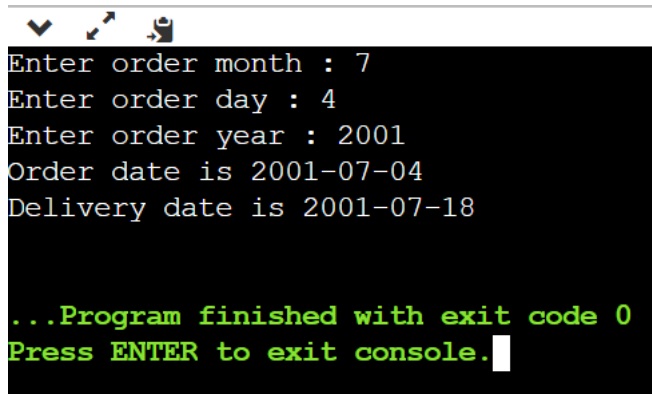
Example program given in reference material for exercise-5 :-

```

/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-2 */
/* Example program given in reference manual for exercise-5 */
import java.time.*;
import java.util.Scanner;
public class DeliveryDate
{
    public static void main(String[] args)
    {
        Scanner input = new Scanner(System.in);
        LocalDate orderDate;
        int mo;
        int day;
        int year;
        final int WEEKS_FOR_DELIVERY = 2;
        System.out.print("Enter order month ");
        mo = input.nextInt();
        System.out.print("Enter order day ");
        day = input.nextInt();
        System.out.print("Enter order year ");
        year = input.nextInt();
        orderDate = LocalDate.of(year, mo, day);
        System.out.println("Order date is " + orderDate);
        System.out.println("Delivery date is " + orderDate.plusWeeks(WEEKS_FOR_DELIVERY)) ;
    }
}

```

Output :-



```

Enter order month : 7
Enter order day : 4
Enter order year : 2001
Order date is 2001-07-04
Delivery date is 2001-07-18

...Program finished with exit code 0
Press ENTER to exit console.

```