EX NO:	PROGRAMS USING CONTROL STRUCTURES AND ARRAYS
DATE :	

Aim

To write program for basic operations using control structures and arrays in java

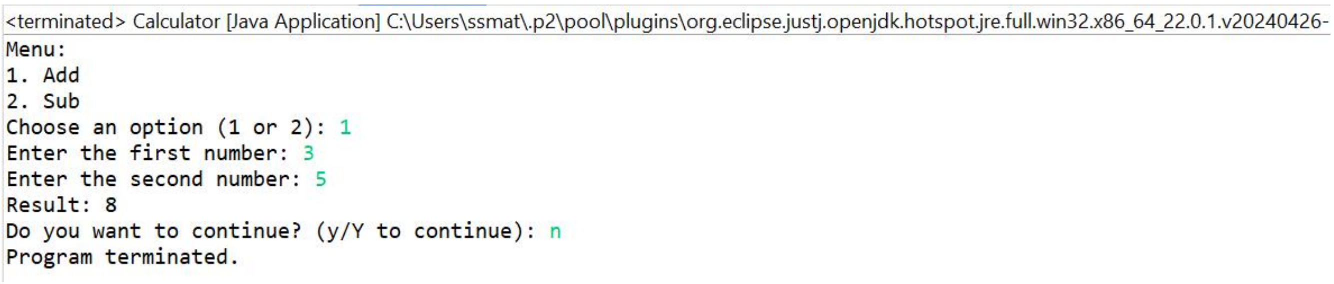
1. Question

Write a program that displays a menu with options 1. Add 2. Sub Based on the options chosen, read 2 numbers and perform the relevant operation. After performing the operation, the program should ask the user if he wants to continue. If the user presses y or Y, then the program should continue displaying the menu else the program should terminate. [Note: Use Scanner class, you can take help from the trainer regarding the same]

Program / Source Code:

```
package dhanush;
import java.util.Scanner;
public class Calculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        char choice;

        do {
            System.out.println("Menu:");
            System.out.println("1. Add");
            System.out.println("2. Sub");
            System.out.print("Choose an option (1 or 2): ");
            int option = scanner.nextInt();
            System.out.print("Enter the first number: ");
            int num1 = scanner.nextInt();
            System.out.print("Enter the second number: ");
            int num2 = scanner.nextInt();
            switch (option) {
                case 1:
                    System.out.println("Result: " + (num1 + num2));
                    break;
                case 2:
                    System.out.println("Result: " + (num1 - num2));
                    break;
                default:
                    System.out.println("Invalid option. Please choose 1 or 2.");
            }

            System.out.print("Do you want to continue? (y/Y to continue): ");
            choice = scanner.next().charAt(0);

        } while (choice == 'y' || choice == 'Y');
        System.out.println("Program terminated.");
        scanner.close();
    }
}
```

Output



```
<terminated> Calculator [Java Application] C:\Users\ssmat\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.0.1.v20240426-
Menu:
1. Add
2. Sub
Choose an option (1 or 2): 1
Enter the first number: 3
Enter the second number: 5
Result: 8
Do you want to continue? (y/Y to continue): n
Program terminated.
```

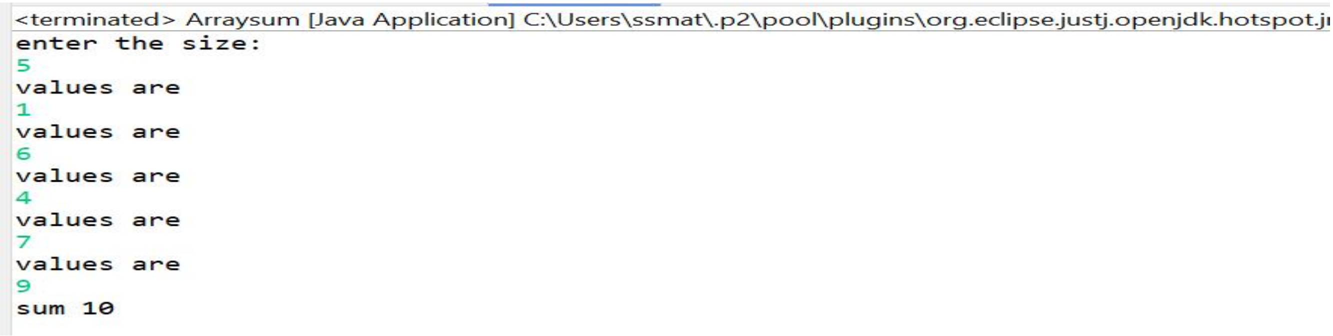
2. Question

Write a program to print the sum of the elements of the array with the given below condition. If the array has 6 and 7 in succeeding orders, ignore 6 and 7 and the numbers between them for the calculation of sum. Eg1) Array Elements - 10,3,6,1,2,7,9 O/P: 22 [i.e 10+3+9] Eg2) Array Elements - 7,1,2,3,6 O/P:19 Eg3) Array Elements - 1,6,4,7,9 O/P:10

Program / Source Code:

```
package dhanush;
import java.util.*;
public class Arraysum {
public static void main (String [] args) {
Scanner sc= new Scanner(System.in);
System.out.println("enter the size:");
int size=sc.nextInt();
int[] arr = new int[size];
for (int i=0; i<size;i++) {
System.out.println("values are");
arr[i]=sc.nextInt();
}
int sum1=0;
for (int i=0; i< arr.length;i++) {
if (arr[i]==6) {
for (int j=i+1;j<arr.length;j++){
if(arr[j]==7) {
i=j+1;
break;
}
}
}
sum1+=arr[i];
}System.out.println("sum "+sum1);
}

}
```

Output



```
<terminated> Arraysum [Java Application] C:\Users\ssmat\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.j
enter the size:
5
values are
1
values are
6
values are
4
values are
7
values are
9
sum 10
```

3. Question

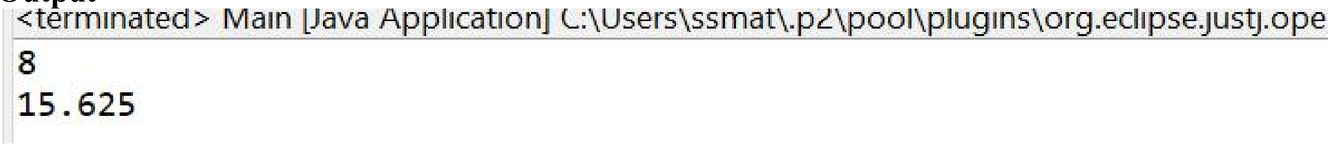
Create a new class called "Calculator" which contains the following:

- A static method called powerInt(int num1,int num2) that accepts two integers and
- returns num1 to the power of num2 (num1 power num2).
- A static method called powerDouble(double num1,int num2) that accepts one double and one integer and returns num1 to the power of num2 (num1 power num2).
- Call your method from another class without instantiating the class (i.e. call it like Calculator.powerInt(12,10) since your methods are defined to be static) Hint: Use Math.pow(double,double) tocalculate the power.

Program / Source Code:

```
package dhanush;
class PowerCalculator {
    public static int powerInt(int num1, int num2) {
        return (int) Math.pow(num1, num2);
    }
    public static double powerDouble(double num1, int num2) {
        return Math.pow(num1, num2);
    }
}
public class Main {
    public static void main(String[] args) {
        int intResult = PowerCalculator.powerInt(2, 3);
        double doubleResult = PowerCalculator.powerDouble(2.5, 3);
        System.out.println(intResult);
        System.out.println(doubleResult);
}
}
```

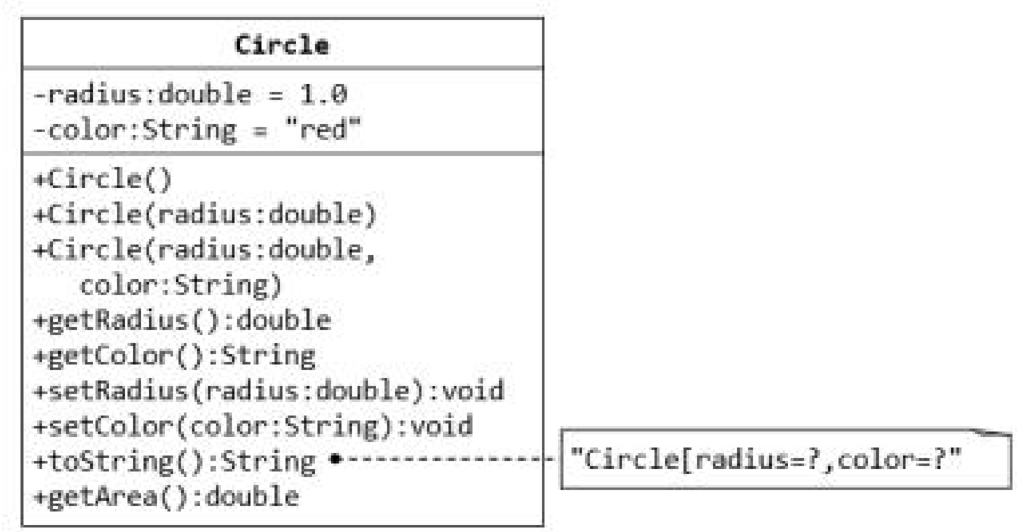
Output



```
<terminated> Main [Java Application] C:\Users\ssmat\.p2\pool\plugins\org.eclipse.justj.ope
8
15.625
```

4. Question

Consider the following UML class diagram**:**



```
Circle
-radius:double = 1.0
-color:String = "red"
+Circle()
+Circle(radius:double)
+Circle(radius:double,
   color:String)
+getRadius():double
+getColor():String
+setRadius(radius:double):void
+setColor(color:String):void
+toString():String
+getArea():double
```

Write a Java class for the Circle class based on the UML class diagram.

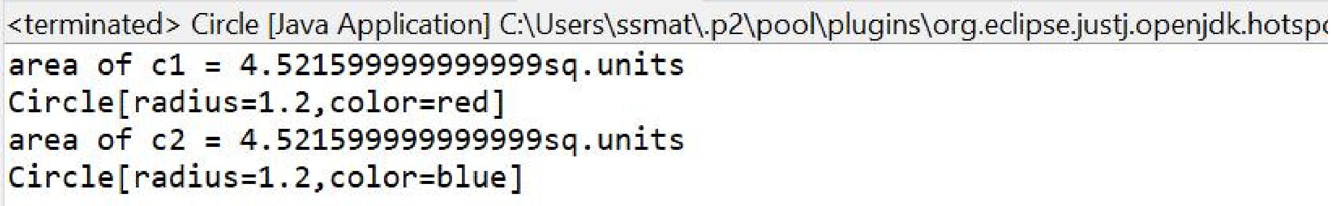
- Write a main class TestCircle that creates two objects for the Circle class and invoke all the methods.

Program / Source Code:

```
public class Circle {
private double radius;
private String color;
public void Circle(){
radius=1.2;
color= "red";
}
public void Circle(double radius,String color){
this.radius=radius;
this.color=color;
}
String getColor(){
return color;
}
double getRadius() {   //get method- have to get the value
return radius;
}
double getArea() {
return 3.14*radius*radius;
}
void setColor(String color) {   //set method-have to assign the value
this.color=color;
}
void setRadius(double radius) {
this.radius=radius;
}
public String toString() {
```

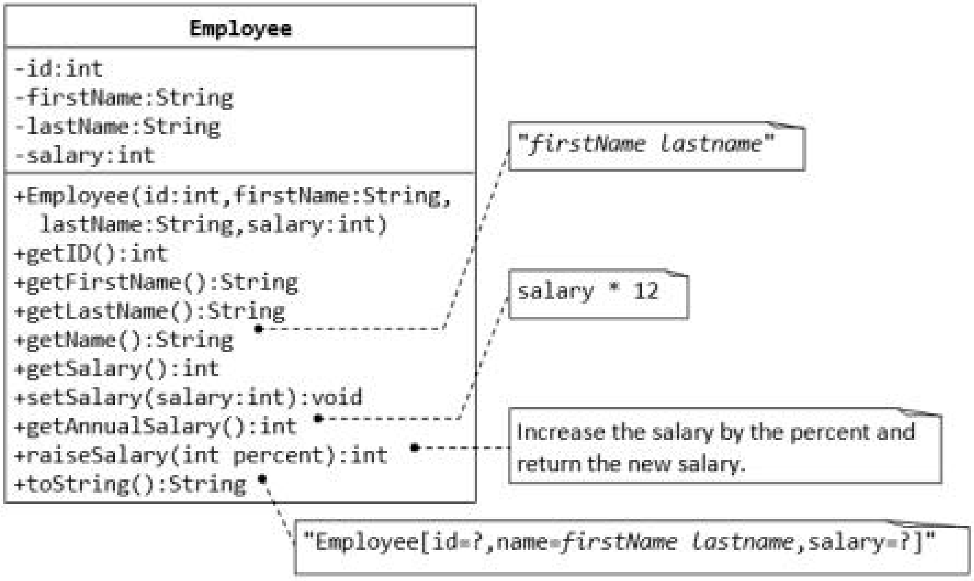
```
return "Circle[radius=" + radius + "," + "color=" + color + "]";
}
}
Public class TestCircle{
public static void main(String[]args) {
Circle c1=new Circle();
Circle c2=new Circle();
c1.setRadius(1.2);
c1.setColor("red");
c1.getRadius();
c1.getColor();
double Area=c1.getArea();
System.out.println("area of c1 = "+ Area +"sq.units");
System.out.println(c1);

c2.setRadius(1.2);
c2.setColor("blue");
c2.getRadius();
c2.getColor();
double area=c2.getArea();
System.out.println("area of c2 = "+ area +"sq.units");
System.out.println(c2);
}
}
```

Output



```
<terminated> Circle [Java Application] C:\Users\ssmat\.p2\pool\plugins\org.eclipse.justj.openjdk.hotsp
area of c1 = 4.521599999999999sq.units
Circle[radius=1.2,color=red]
area of c2 = 4.521599999999999sq.units
Circle[radius=1.2,color=blue]
```

5. Question


Employee
-id:int
-firstName:String
-lastName:String
-salary:int
+Employee(id:int,firstName:String,
lastName:String,salary:int)
+getID():int
+getFirstName():String
+getLastName():String
+getName():String
+getSalary():int
+setSalary(salary:int):void
+getAnnualSalary():int
+raiseSalary(int percent):int
+toString():String
"firstName lastname"
salary * 12
Increase the salary by the percent and return the new salary.
"Employee[id=?,name=firstName lastname,salary=?]"

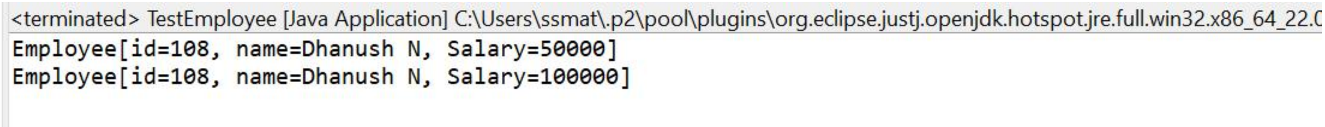
- Write a Java class for the Employee class based on the UML class diagram.
- Write a main class TestEmployee that creates two objects for the Employee class and invoke all the methods.

Program / Source Code:

```
package dhanush;

public class Employee {
        private int id;
        private String firstname;
        private String lastname;
        private int salary;
        Employee(int id ,String firstname,String lastname,int salary){
                this.id=id;
                this.firstname=firstname;
                this.lastname=lastname;
                this.salary=salary;
        }
        public int getId() {
        return id;
        }
        public String getFirstName()
        {
                return firstname;
        }
        public String getLastName()
        {
                return lastname;}
        public String getName() {
                return firstname+" "+lastname;
        }
   public int getSalary() {
      return salary;
   }
   public void setSalary(int salary) {
      this.salary = salary;
   }
   public int getAnnualSalary() {
      return salary * 12;
   }
   public int raiseSalary() {
        return salary * 2/100;
   }
   public String toString() {
      return "Employee[id=" + id + ", name=" + firstname + " " + lastname + ", Salary=" + salary + "]";
        }
}

public class TestEmployee {
   public static void main(String[] args) {
      Employee e1 = new Employee(108,"Dhanush","N",60000);
```

```
        e1.setSalary(50000);
        e1.getAnnualSalary();
        e1.raiseSalary();
        System.out.println(e1);

        e1.setSalary(100000);
        System.out.println(e1);
    }
}
```

Output



```
<terminated> TestEmployee [Java Application] C:\Users\ssmat\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_22.C
Employee[id=108, name=Dhanush N, Salary=50000]
Employee[id=108, name=Dhanush N, Salary=100000]
```

Result

Thus, the java program was executed successfully and the output was verified.