

Name: Pallavi Chaudhary

Student Code: AF0316472

Batch code: ANP-C6008

Lab Assignment 5

Write a Java program to create a class called Shape with methods called `getPerimeter()` and `getArea()`. Create a subclass called Circle that overrides the `getPerimeter()` and `getArea()` methods to calculate the area and perimeter of a circle.

Code: -

```
//1)Write a Java program to create a
//class called
//Shape with
//methods called
//
//getPerimeter()
//
//and getArea(). Create a subclass called Circle that overrides the getPerimeter()
//
//and getArea() methods to calculate the area and perimeter of a circle.

package Overriding;

class Shape { //super class
    void getPerimeter(int r) { //method
        System.out.println("The parimeter : "); //statement
```

```
}
```

```
void getArea(int r) { //method
```

```
    System.out.println("The area : "); //statement
```

```
}
```

```
}
```

```
public class Circle extends Shape { //subclass
```

```
    @Override
```

```
void getPerimeter(int r) { //method
```

```
    System.out.println(2 * 3.14 * r); //statement
```

```
}
```

```
void getArea(int r) { //method
```

```
    System.out.println(3.14 * r * r); //statement
```

```
}
```

```
public static void main(String[] args) { //main method
```

```
    Circle c = new Circle(); //object of subclass c
```

```
    System.out.println("The parimeter of circle : "); //Statement
```

```
    c.getPerimeter(3); //calling the method by object
```

```

        System.out.println("=====
=====");

```

```

        System.out.println("The area of circle : "); //Statement

```

```

        c.getArea(3); //calling the method

```

```

    }

}

```

Output: -

```

The parimeter of circle :
18.84
=====
The area of circle :
28.259999999999998

```

2)Write a program to search an element in a given array by using Binary Search method without using predefined method.

Code: -

```

package Assignment5;

public class pro2 {

    public static void main(String[] args) {
        int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
        int f = 0; // f--> first index
        int l = 9; // l--> last index
        int key = 8;
        //System.out.println("Array : "+a);

        while (a[f] <= a[l]) { //a[0]<= a[9]-->1<=10-->true,
a[5]<=a[9]-->6<=10-->true, a[6]=7<10-->true
            int mid = (f + l) / 2; // mid--> middle index

            if (a[mid] < key) //a[4]= 5<8 --> true, a[7]=
8<8-->false, a[

```

```

        f = mid + 1; // f = 4+1 = 5 , f = 5+1=6

        else if (a[mid] == key) { //a[4] = 8--> false,
a[7] = 8 = key = 8
        System.out.println("Element found at index
: "+mid); //statement wouldn't pass
        break;}
        else {
            l = mid - 1;}
    }
}

```

Output: -

Element found at index : 7

3) Create an interface called "BankAccount" with methods called "deposit" and "withdraw". Create a class called "CheckingAccount" that implements the BankAccount interface and implements the "deposit" and "withdraw" methods. Create an object of the CheckingAccount class and call both the "deposit" and "withdraw" methods.

Code: -

```

//3) Create an interface called "BankAccount" with methods
called "deposit" and "withdraw".
//Create a class called "CheckingAccount" that implements the
BankAccount
//interface and implements the "deposit" and "withdraw"
methods. Create an object of
//the CheckingAccount class and call both the "deposit" and
"withdraw" methods.

```

```

package Assignment5;

```

```

interface BankAccount { //BankAccount interface
    void deposit(); //method

    void withdraw(); //method
}

```

```

class CheckingAccount implements BankAccount {
//implementation

```

```
        public void deposit() { //implementing the method
            System.out.println("deposit the amount");
        }

        public void withdraw() { //implementing another method
            System.out.println("withdraw the amount");
        }

        public static void main(String[] args) { //calling main
            method
            CheckingAccount ca = new CheckingAccount();
            //creating object of CheckingAccount class
            ca.deposit(); //calling method by object
            ca.withdraw(); //calling method by object
        }
    }
```

Output: -

```
deposit the amount
withdraw the amount
|
```