TECHNO MAIN SALT LAKE

LAB REPORT ON

PAPER – MCAN293

OOPs Programming Lab using JAVA

MCA (Masters Of Computer Application)

SEMESTER – II

College Code – 130

NAME : *KOUSHIK MANDAL*

UNIVERSITY ROLL NO : *13071022029*

UNIVERSITY REGISTRATION NO : *221300510023*

STUDENT ID : *2213026015*

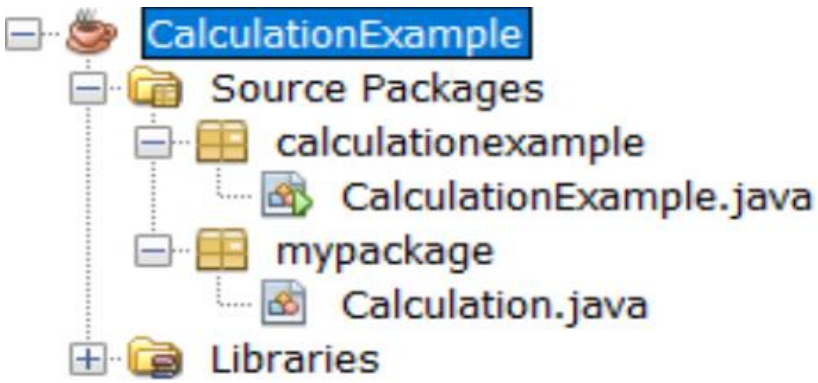
SESSION : *2022-2023*

Assigment 1

1.

❖ Source Code :

- **File Structure :**



- **Calculation.java :**

```
package mypackage;

import java.math.BigInteger;

public class Calculation {
    // function to calculate the factorial value of a number
    public static BigInteger factorial(int num) {
        BigInteger fact = BigInteger.valueOf(1);
        for(int i = 1; i <= num; i++) {
            fact = fact.multiply(BigInteger.valueOf(i));
        }
        return fact;
    }

    // function to check if a number is prime or not
    public static boolean isPrime(long n) {
        if(n <= 1) {
            return false;
        }
        for(long i = 2; i*i <= n; i++) {
            if(n % i == 0) {
                return false;
            }
        }
        return true;
    }

    // function to find the next probable prime number
```

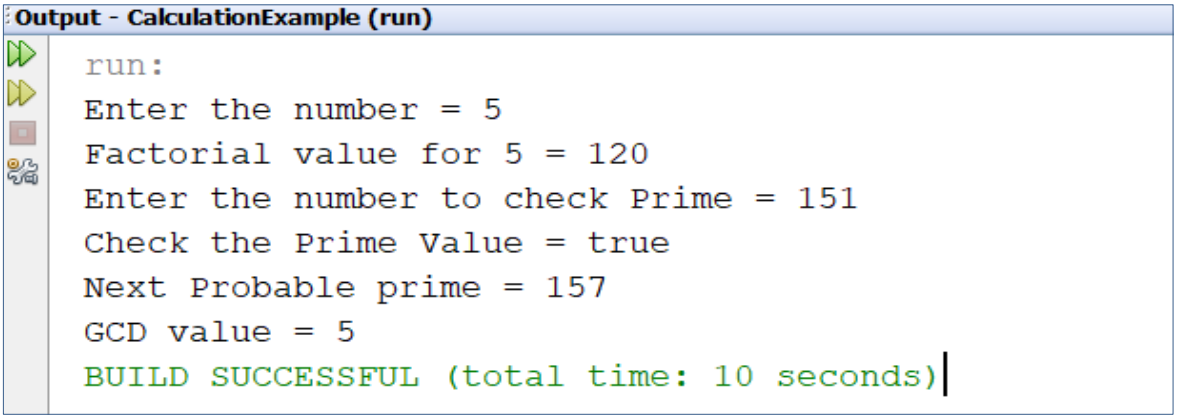
```
    public static BigInteger nextPrime(long n) {
        BigInteger num = BigInteger.valueOf(n+1);
        while(!num.isProbablePrime(100)) {
            num = num.add(BigInteger.ONE);
        }
        return num;
    }

    // function to calculate the gcd of two numbers
    public static int gcdCalculation(int num1, int num2) {
        while(num2 != 0) {
            int temp = num1 % num2;
            num1 = num2;
            num2 = temp;
        }
        return num1;
    }

    // function to calculate the gcd of two big integer numbers
    public static BigInteger gcdCalculation(BigInteger num1,
BigInteger num2) {
        return num1.gcd(num2);
    }
}
```

- **CalculationExample.java :**

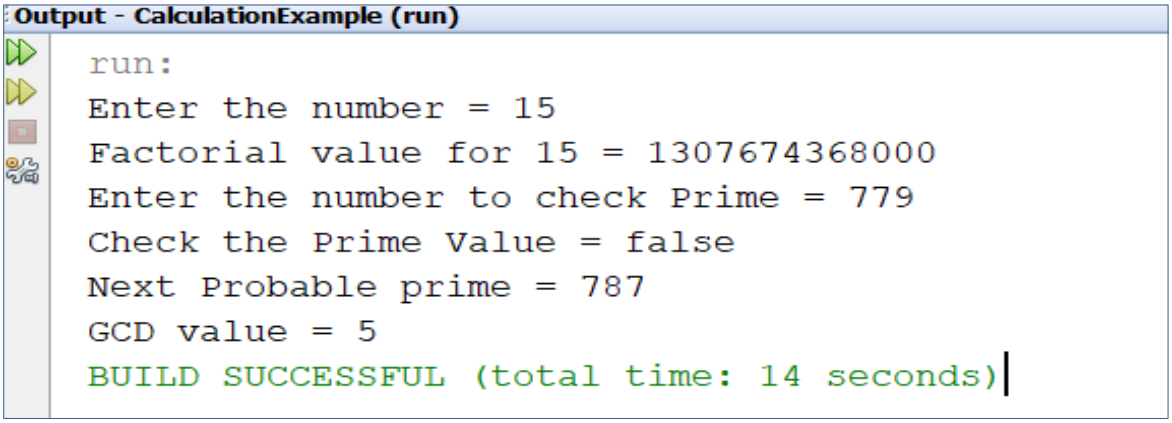
```
package calculationexample;
import mypackage.Calculation;
import java.util.Scanner;
import java.math.BigInteger;

public class CalculationExample {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Calculation cal = new Calculation();

        System.out.print("Enter the number = ");
        int num = sc.nextInt();
        BigInteger factorialValue = cal.factorial(num);
        System.out.println("Factorial value for " + num + " = " +
factorialValue);

        System.out.print("Enter the number to check Prime = ");
        long n = sc.nextLong();
        boolean val = cal.isPrime(n);
        System.out.println("Check the Prime Value = " + val);
        BigInteger nextPrime = cal.nextPrime(n);
        System.out.println("Next Probable prime = " + nextPrime);

        System.out.println("GCD value = " +
cal.gcdCalculation(6786540, 4587655));
    }
}
```

❖ Output :

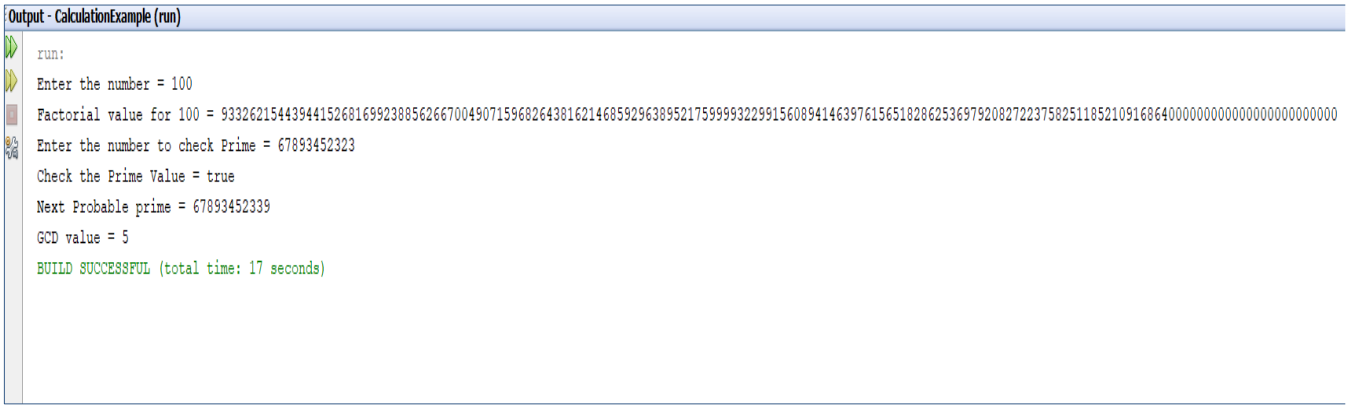
- **Case 1 :**



- **Case 2 :**



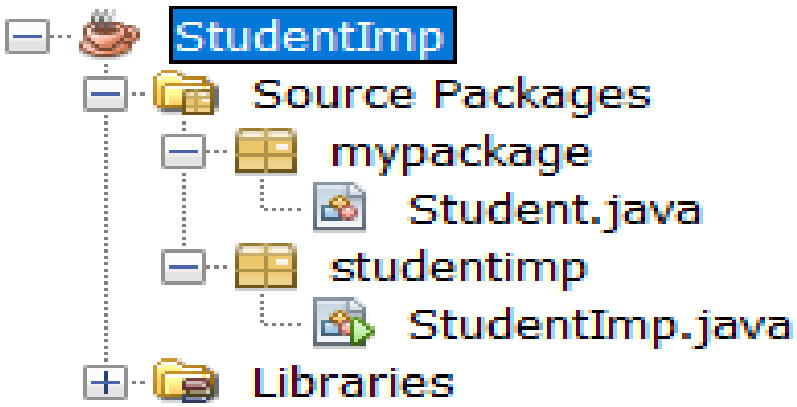
- **Case 3 :**



2.

❖ Source Code :

- **File Structure :**



- **Student.java :**

```
package mypackage;

public class Student {
    private String stName_;
    private int stroll_;
    private int stMarks_;
    private String stYear_;

    public Student() {
        stroll_ = 0;
        stMarks_ = -1;
    }

    public Student(String name, int roll, int marks, String year) {
        this();
        setStName(name);
        setStRoll(roll);
        setStMarks(marks);
        setStYear(year);
    }

    public String getStName() {
        return stName_;
    }

    public void setStName(String stName) {
        this.stName_ = stName;
    }

    public int getStRoll() {
        return stroll_;
    }
```

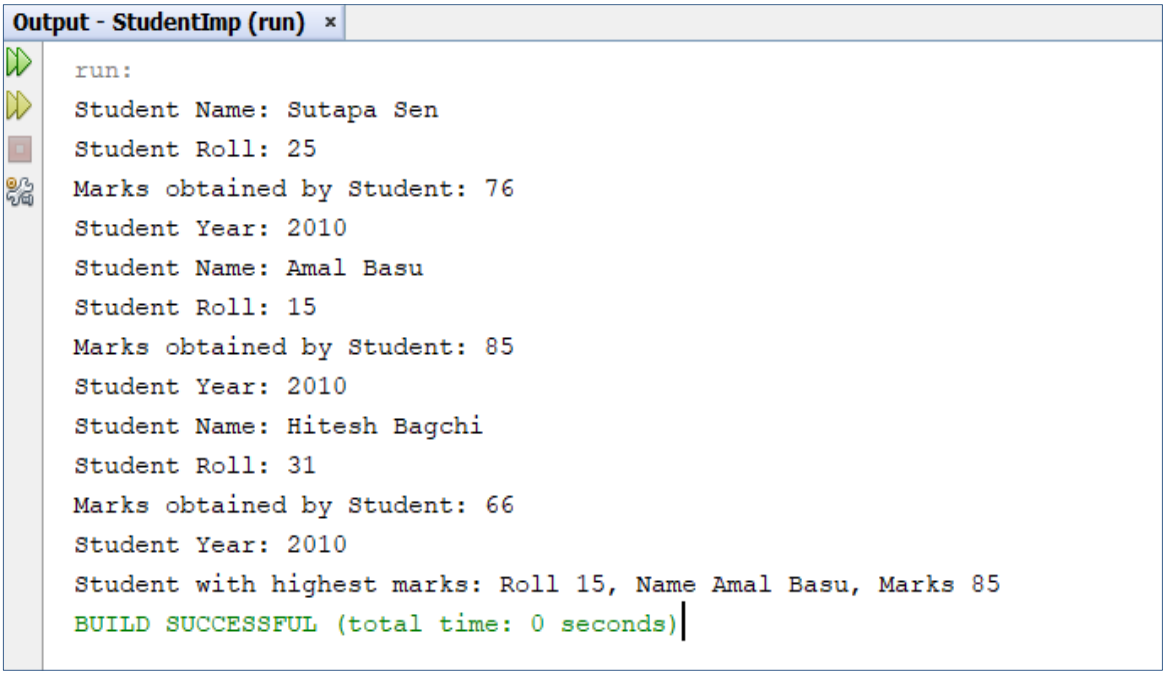
```
    public void setStRoll(int stroll) {
        if (stroll >= 1 && stroll <= 60) {
            this.stroll_ = stroll;
        } else {
            System.out.println("Invalid roll number.");
        }
    }

    public int getStMarks() {
        return stMarks_;
    }

    public void setStMarks(int stMarks) {
        if (stMarks >= 0 && stMarks <= 100) {
            this.stMarks_ = stMarks;
        } else {
            System.out.println("Invalid marks.");
        }
    }

    public String getStYear() {
        return stYear_;
    }

    public void setStYear(String stYear) {
        this.stYear_ = stYear;
    }

    public void display() {
        System.out.println("Student Name: " + getStName());
        System.out.println("Student Roll: " + getStRoll());
        System.out.println("Marks obtained by Student: " +
getStMarks());
        System.out.println("Student Year: " + getStYear());
    }

    public Student displayRollName() {
        System.out.println("Student Roll: " + getStRoll() + ", Name:
" + getStName());
        return this;
    }
}
```

- **StudentImp.java :**

```
package studentimp;
import mypackage.Student;

public class StudentImp {
    public static void main(String[] args) {
        Student s1 = new Student("Sutapa Sen", 25, 76, "2010");
        Student s2 = new Student("Amal Basu", 15, 85, "2010");
        Student s3 = new Student("Hitesh Bagchi", 31, 66, "2010");

        s1.display();
        s2.display();
        s3.display();
```

```
        Student[] students = {s1, s2, s3};
        int maxMarks = -1;
        Student topStudent = null;
        for (Student s : students) {
            if (s.getStMarks() > maxMarks) {
                maxMarks = s.getStMarks();
                topStudent = s;
            }
        }
        if (topStudent != null) {
            System.out.println("Student with highest marks: Roll " +
topStudent.getStRoll() +
                    ", Name " + topStudent.getStName() + ", Marks " +
topStudent.getStMarks());
        }
    }
}
```

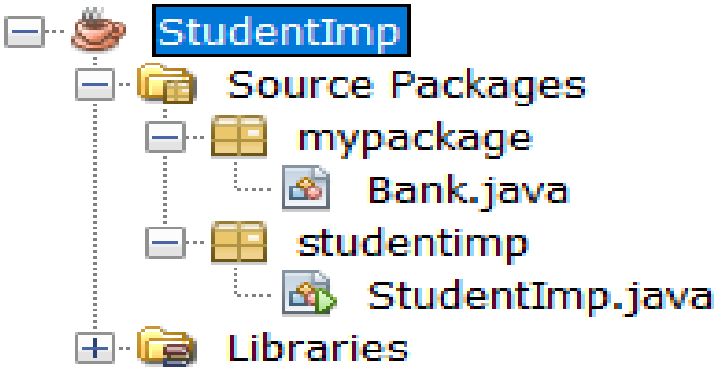
❖ Output :



3.

❖ Source Code :

- **File Structure :**



- **Bank.java :**

```
package mypackage;

public class Bank {
    private String accountNumber;
    private String customerName;
    private char accountType;
    private String balanceAmount;

    public Bank(String accountNumber, String customerName, char
accountType, String balanceAmount) {
        this.accountNumber = accountNumber;
        this.customerName = customerName;
        this.accountType = accountType;
        this.balanceAmount = balanceAmount;
    }

    public void display() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance Amount: " + balanceAmount);
    }

    public void deposit(String amount) {
        double depositAmount = Double.parseDouble(amount);
        double currentBalance = Double.parseDouble(balanceAmount);
        double updatedBalance = currentBalance + depositAmount;
        balanceAmount = String.format("%.2f", updatedBalance);
    }

    public void withDraw(String amount) {
        double withdrawalAmount = Double.parseDouble(amount);
        double currentBalance = Double.parseDouble(balanceAmount);
```

```
        if (currentBalance - withdrawalAmount >= 5000) {
            double updatedBalance = currentBalance -
withdrawalAmount;
            balanceAmount = String.format("%.2f", updatedBalance);
        } else {
            System.out.println("Insufficient balance");
        }
    }
}
```

- **StudentImp.java :**

```
package studentimp;
import mypackage.Bank;

public class StudentImp {
    public static void main(String[] args) {
        Bank B1 = new Bank("01634001300000452","Biman Sen", 'S',
                "55000.67");
        B1.display();
        System.out.println("After Deposit ...");
        B1.deposit("45987.59");
        B1.display();
        System.out.println("After With draw ...");
        B1.withDraw("10000");
        B1.display();
    }//end of main
}//end of class
```

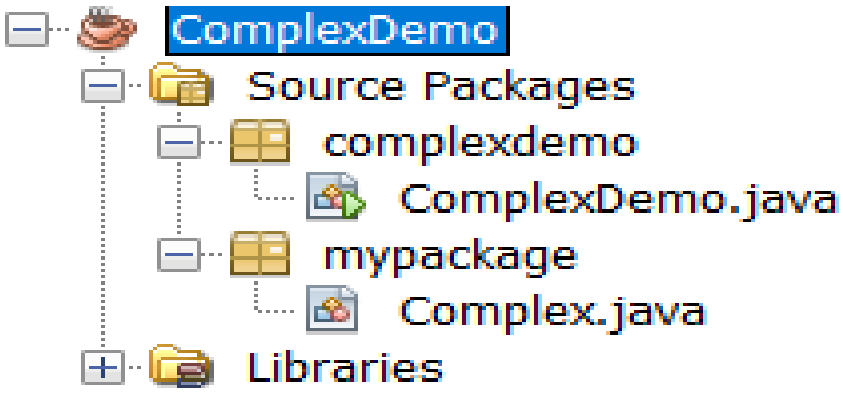
❖ Output :

```
Output - StudentImp (run)
run:
Account Number: 01634001300000452
Customer Name: Biman Sen
Account Type: S
Balance Amount: 55000.67
After Deposit ...
Account Number: 01634001300000452
Customer Name: Biman Sen
Account Type: S
Balance Amount: 100988.26
After With draw ...
Account Number: 01634001300000452
Customer Name: Biman Sen
Account Type: S
Balance Amount: 90988.26
BUILD SUCCESSFUL (total time: 0 seconds)
```

4.

❖ Source Code :

- **File Structure :**



- **Complex.java :**

```
package mypackage;

public class Complex {
    private double realPart;
    private double imaginaryPart;

    // Default constructor
    public Complex() {
        this.realPart = 0;
        this.imaginaryPart = 0;
    }

    // Constructor with parameters
    public Complex(double realPart, double imaginaryPart) {
        this.realPart = realPart;
        this.imaginaryPart = imaginaryPart;
    }

    // Method to add two complex numbers
    public Complex add(Complex otherNumber) {
        double real = this.realPart + otherNumber.realPart;
        double imaginary = this.imaginaryPart +
otherNumber.imaginaryPart;
        return new Complex(real, imaginary);
    }

    // Method to subtract two complex numbers
    public Complex subtract(Complex otherNumber) {
        double real = this.realPart - otherNumber.realPart;
        double imaginary = this.imaginaryPart -
otherNumber.imaginaryPart;
        return new Complex(real, imaginary);
    }
```

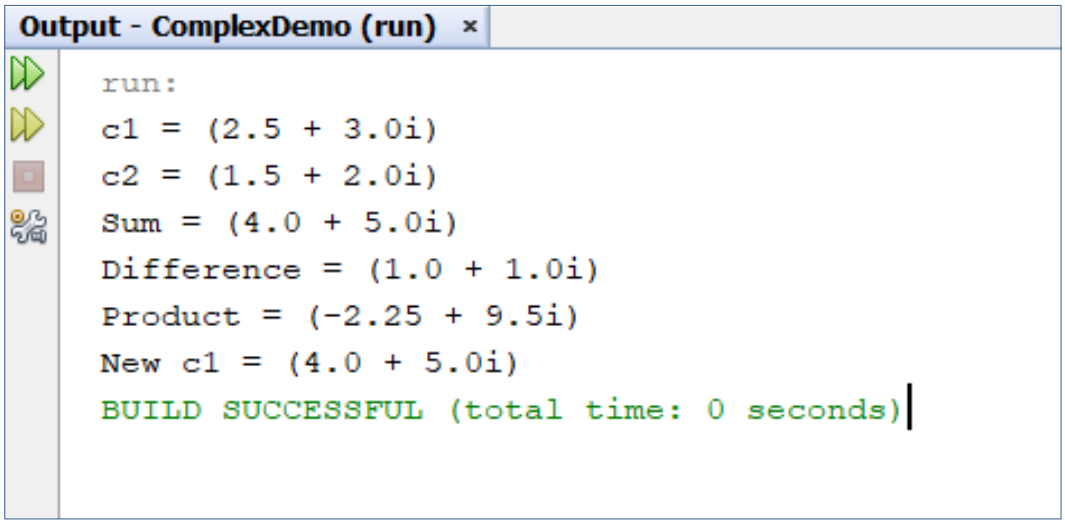
```
    // Method to multiply two complex numbers
    public Complex multiply(Complex otherNumber) {
        double real = this.realPart * otherNumber.realPart -
this.imaginaryPart * otherNumber.imaginaryPart;
        double imaginary = this.realPart * otherNumber.imaginaryPart
+ this.imaginaryPart * otherNumber.realPart;
        return new Complex(real, imaginary);
    }

    // Setter method for realPart
    public void setRealPart(double realPart) {
        this.realPart = realPart;
    }

    // Setter method for imaginaryPart
    public void setImaginaryPart(double imaginaryPart) {
        this.imaginaryPart = imaginaryPart;
    }

    // Getter method for realPart
    public double getRealPart() {
        return realPart;
    }

    // Getter method for imaginaryPart
    public double getImaginaryPart() {
        return imaginaryPart;
    }

    // toString method to print complex number
    public String toString() {
        return "(" + realPart + " + " + imaginaryPart + "i)";
    }
}
```

- **ComplexDemo.java :**

```
package complexdemo;
import mypackage.Complex;


public class ComplexDemo {
    public static void main(String[] args) {
        // Creating two complex numbers
        Complex c1 = new Complex(2.5, 3.0);
        Complex c2 = new Complex(1.5, 2.0);

        // Printing the complex numbers
        System.out.println("c1 = " + c1.toString());
        System.out.println("c2 = " + c2.toString());

        // Adding the two complex numbers
        Complex sum = c1.add(c2);
        System.out.println("Sum = " + sum.toString());
```

```
        // Subtracting the two complex numbers
        Complex difference = c1.subtract(c2);
        System.out.println("Difference = " + difference.toString());

        // Multiplying the two complex numbers
        Complex product = c1.multiply(c2);
        System.out.println("Product = " + product.toString());

        // Setting new values for realPart and imaginaryPart of c1
        c1.setRealPart(4.0);
        c1.setImaginaryPart(5.0);

        // Printing the new values of c1
        System.out.println("New c1 = " + c1.toString());
    }
}
```

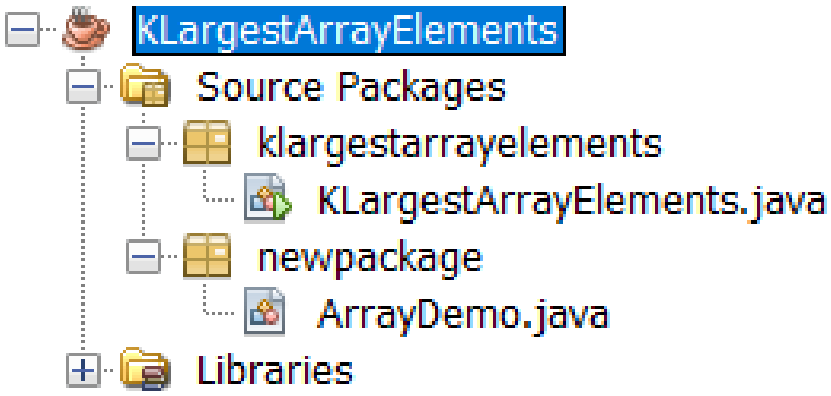
❖ Output :



5.

❖ Source Code :

- **File Structure :**



- **ArrayDemo.java :**

```
package newpackage;

import java.util.Arrays;

public class ArrayDemo {

    int[] arr;

    public ArrayDemo(int[] arr) {
        this.arr = arr;
    }

    public static int[] findKLargest(int[] arr, int k) {
        Arrays.sort(arr);
        int[] kLargest = new int[k];

        int index = 0;
        for (int i = arr.length - k; i < arr.length; i++) {
            kLargest[index++] = arr[i];
        }

        return kLargest;
    }

}
```

- **KLargestArrayElements.java :**

```
package klargestarrayelements;
import newpackage.ArrayDemo;

import java.util.Arrays;

public class KLargestArrayElements {
    public static void main(String[] args) {

        int[] arr = { 12, 45, 1, -1, 45, 16, 97, 100 };
        int k = 3;

        ArrayDemo obj = new ArrayDemo(arr);

        int[] kLargest = obj.findKLargest(arr, k);
        System.out.println("The " + k + " largest elements in the
array are: " + Arrays.toString(kLargest));
    }
}
```

❖ Output :