Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 **Section: CSE 213 - DA (PC)**

Final Lab Report (Portfolio)

Student Details

Name	ID
Md. Shahidul Islam Prodhan	**213902017**

Submission Date: 01 January 2023

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 01

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 **Section: CSE 213 - DA (PC)**

Lab Experiment Name(s):

- Implement checking of odd and even number
- Implement summation of factorial odd number series.

$$\text{Sum} = \frac{x^2}{1!} + \frac{x^4}{3!} + \frac{x^6}{5!} + \dots \frac{x^n}{(n-1)!}$$

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 17 October, 2022

Submission Date: 23 October, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Lab Report of C - Java Syntax Similarity : Array, Conditionals, Loops

2. OBJECTIVES

• To gather knowledge of java syntax, array conditions and loops.
• To implement different types of problem like prime number, Fibonacci number, odd even number checking
solved in Java.

3. PROCEDURE/ ANALYSIS / DESIGN

Problem 1: Implement checking of odd and even number

1) Start
2) Create an object of the Scanner class to take input from the user.
3) Declare a variable to store the number.
4) Ask the user to initialize the number.
5) Check whether the number is even or odd by using bitwise XOR.
6) If the number after bitwise XOR with 1 is equal to the original number + 1, then it is an even number.
7) If not equal, then it is an odd number.
8) Display the result.
9) Stop.

Problem 2: Implement summation of factorial odd number series.

1)Use Scanner for user take input on X & N value.
2)Use for loop.
3)Sum= X to the power devided N odd Series factorial.
4)Sum = sum+ value

Implementation

1)Use netbeans Application.
2)Create a project name lab_report_1_problem_solve
3)Take scanner faction foe user take input
4)User giving X &N value this value store x & n
5)Use for loop 0 to n
6)Find odd series factorial & X to the power even series.
7)At last sum = X to the power / factorial and print the sum value.

4. IMPLEMENTATION & TEST RESULT

Problem 1: Implement checking of odd and even number

Main.java

```java
import java.util.Scanner;

public class EvenOdd {

    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int num = reader.nextInt();

        if(num % 2 == 0)
            System.out.println(num + " is an even number.");
        else
            System.out.println(num + " is an odd number.");
    }
}
```

Output

```
java -cp /tmp/apmjOORQ9w EvenOdd
Enter a number: 22
22 is an even number.
```

4. IMPLEMENTATION & TEST RESULT

Problem 2: Implement summation of factorial odd number series.

```
public class SeriesSum
    {
    int x,n;
    double sum;
    SeriesSum(int xx,int nn)
    { x=xx;
    n=nn;
    sum=0.0;
    }
    double findfact(int a)
    { return (a<2)? 1:a*findfact(a-1);
    }
    double findpower(int a, int b)
    { return (b==0)? 1:a*findpower(a,b-1);
    }
    void calculate()
    {
    System.out.println("x ="+x);
    System.out.println("n ="+n);
    for(int i=2;i<=n;i+=2){

        System.out.println(x+"^"+i+"/"+(i-1)+"!" +" = " +(findpower(x,i)+"/"+findfact(i-1)) );
        //System.out.println(findpower(x,i)+"/"+findfact(i-1));

       sum += findpower(x,i)/findfact(i-1);
    }
    }
    void display()
    { System.out.println("sum="+ sum);
    }
    public static void main(String arg[])
    { SeriesSum obj = new SeriesSum(3,9);
    obj.calculate();
    obj.display();
    }
}
```

Result

compiled and executed in 1.516 sec(s)

```
x =3
n =9
3^2/1! = 9.0/1.0
3^4/3! = 81.0/6.0
3^6/5! = 729.0/120.0
3^8/7! = 6561.0/5040.0
sum=29.876785714285713
```

6. ANALYSIS AND DISCUSSION

1) The first problem was quite easy for us to solve, as we have learned C programming and solved such problems using various conditions and loops.
2) The 2nd problem is solved by using Java. The problem was quite hard in the beginning to think out and how to solve this in efficient way. In this program we implement calculations in more efficient way to understand the deeper knowledge of such mathematical problems.

7. SUMMARY

1) We have used basics of java using various conditions and loops which we have performed in C language.
2) We have learned to solve complex mathematical problems from the 2nd problem.

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 02

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 Section: CSE 213 - DA (PC)

Lab Experiment Name(s):

Take three constructor where first constructor will calculate the area of triangle, second constructor will calculate the area of rectangle and third constructor will calculate the area of circle using overloading constructor However, Input must be taken from users.

- Implement the above problem using switch case statements.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 24 October, 2022

Submission Date: 30 October, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Lab Report of Class, Objects, Object Arrays, Constructors, Methods

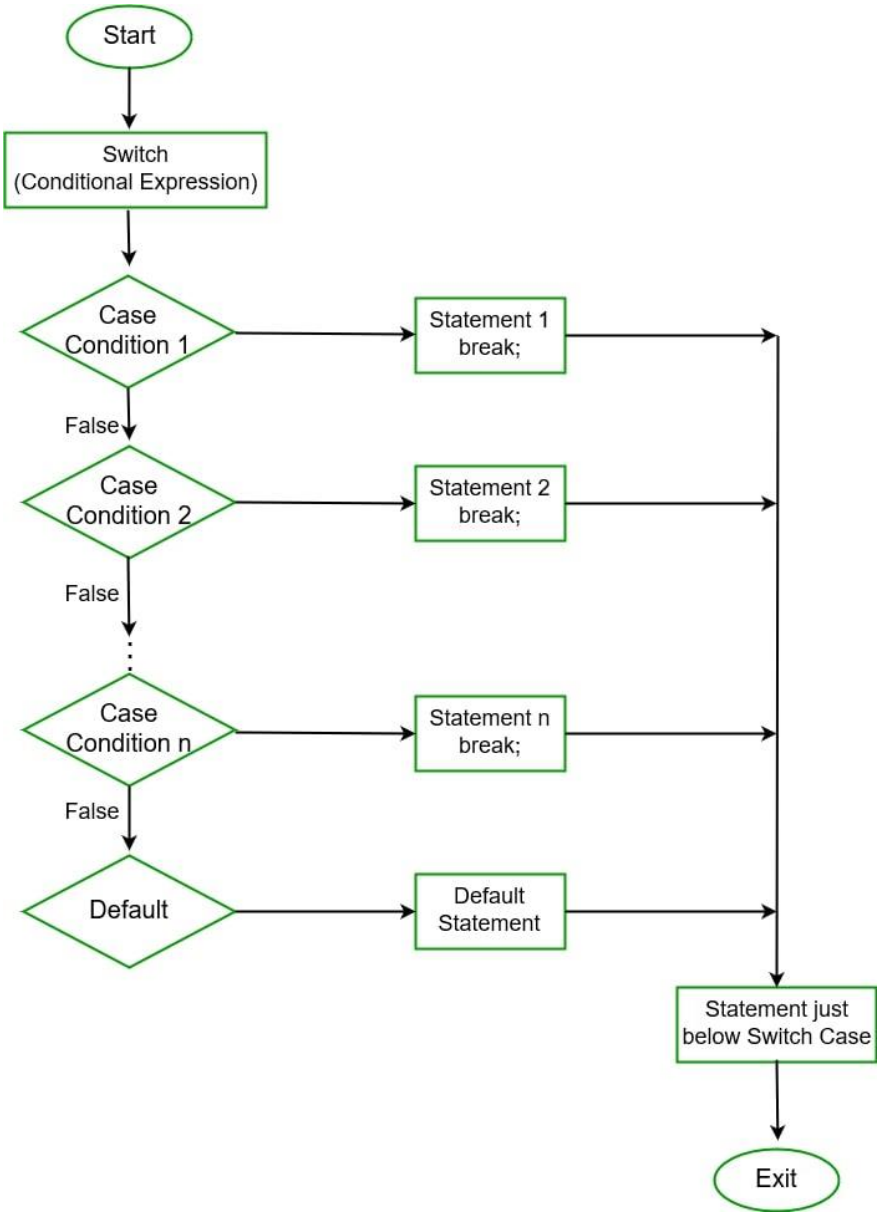
2. OBJECTIVES

- To gather knowledge of Class, Objects, Object Arrays, Constructors, Methods.
- To implement the constructor, array and methods

3. PROCEDURE/ ANALYSIS / DESIGN

Problem 1: Take three constructor where first constructor will calculate the area of triangle, second constructor will calculate the area of rectangle and third constructor will calculate the area of circle using overloading constructor However, Input must be taken from users.

Implement the above problem using switch case statements.



Syntax / Pseudo Code:

```
// switch statement
switch(expression)
{
  // case statements
  // values must be of same type of expression
  case value1 :
    // Statements
    break; // break is optional

  case value2 :
    // Statements
    break; // break is optional

  // We can have any number of case statements
  // below is default statement, used when none of the cases is
true.
  // No break is needed in the default case.
  default :
    // Statements
}
```

4. IMPLEMENTATION & TEST RESULT

Problem 1: Take three constructor where first constructor will calculate the area of triangle, second constructor will calculate the area of rectangle and third constructor will calculate the area of circle using overloading constructor However, Input must be taken from users.

- ***Implement the above problem using switch case statements.***

Main.java

```java
import java.util.Scanner;
public class FindAreaUsingSwitchStatement
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("MENU:");
        System.out.println("");
        System.out.println("1.Area of a circle");
        System.out.println("2.Area of a triangle");
        System.out.println("3.Area of a rectangle");
        System.out.println("");
        System.out.println("Please enter any of the above option: ");
        int num = sc.nextInt();
        switch(num)
        {
            case 1: System.out.println("Please enter radius of circle: ");
            double radius = sc.nextFloat();
            double areaCircle = (22 * radius * radius) / 7;
            System.out.println("Area of circle is: " + areaCircle);
            break;
            case 2: System.out.println("Please enter base and height of triangle: ");
            double base = sc.nextFloat();
            double height = sc.nextFloat();
            double areaTriangle = (base* height) / 2;
            System.out.println("Area of triangle is: " + areaTriangle);
            break;
            case 3: System.out.println("Please enter length and breadth of rectangle: ");
            int length = sc.nextInt();
            int breadth = sc.nextInt();
            int areaRectangle = length * breadth;
            System.out.println("Area of ractangle is: " + areaRectangle);
            break;
            default:System.exit(0);
        }
        sc.close();
    }
}
```

4. IMPLEMENTATION & TEST RESULT

Problem 1: Take three constructor where first constructor will calculate the area of triangle, second constructor will calculate the area of rectangle and third constructor will calculate the area of circle using overloading constructor However, Input must be taken from users.

- ***Implement the above problem using switch case statements.***

Output

```
java -cp /tmp/k7eTL9cOp6 FindAreaUsingSwitchStatement
MENU:

1.Area of a circle
2.Area of a triangle
3.Area of a rectangle

Please enter any of the above option:
1
Please enter radius of circle:
6
Area of circle is: 113.14285714285714
```

Output

```
java -cp /tmp/k7eTL9cOp6 FindAreaUsingSwitchStatement
MENU:
1.Area of a circle
2.Area of a triangle
3.Area of a rectangle

Please enter any of the above option:
2
Please enter base and height of triangle:
4 7
Area of triangle is: 14.0
```

Output

```
java -cp /tmp/k7eTL9cOp6 FindAreaUsingSwitchStatement
MENU:
1.Area of a circle
2.Area of a triangle
3.Area of a rectangle

Please enter any of the above option:
3
Please enter length and breadth of rectangle:
7 8
Area of ractangle is: 56
```

6. ANALYSIS AND DISCUSSION

1) The problem is solved by using Java. The problem was quite hard in the beginning to think out and how to solve this in efficient way. In this program we implement calculations in more efficient way to understand the deeper knowledge of such mathematical problems like using switch case to calculate the area of various things such as Rectangle, Circle and Triangle.

7. SUMMARY

1) We have used basics of java using switch case to calculate the area of various things .
2) We have learned to solve complex mathematical problems from it.

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 03

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 Section: CSE 213 - DA (PC)

Lab Experiment Name(s):

- 2 files contains 2 matrix, read from them and provide the matrix multiplication in the 3rd file.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 31 October, 2022

Submission Date: 06 November, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Lab Report of Package, String, File

2. OBJECTIVES

- Understanding Package
- Introducing String operations
- Implementing common FILE operations in Java

3. PROCEDURE/ ANALYSIS / DESIGN

Problem: 2 files contains 2 matrix, read from them and provide the matrix multiplication in the 3rd file.

Main.java

```java
// Java program to multiply two matrices.
import java.io.*;
import java.util.*;

class GFG{

static int MAX = 100;

// Function to print Matrix
static void printMatrix(int M[][], int rowSize,
                        int colSize)
{
    for(int i = 0; i < rowSize; i++)
    {
        for(int j = 0; j < colSize; j++)
            System.out.print(M[i][j] + " ");

        System.out.println();
    }
}

// Function to multiply two matrices A[][] and B[][]
static void multiplyMatrix(int row1, int col1,
                        int A[][], int row2,
                        int col2, int B[][])
{
    int i, j, k;

    // Matrix to store the result
    int C[][] = new int[MAX][MAX];

    // Check if multiplication is Possible
    if (row2 != col1)
```

Main.java

```java
    // Read size of Matrix A from user
    System.out.print("Enter the number of " +
                     "rows of First Matrix: ");
    row1 = read.nextInt();
    System.out.println(row1);
    System.out.print("Enter the number of " +
                     "columns of First Matrix: ");
    col1 = read.nextInt();
    System.out.println(col1);

    // Read the elements of Matrix A from user
    System.out.println("Enter the elements " +
                     "of First Matrix: ");
    for(i = 0; i < row1; i++)
    {
        for(j = 0; j < col1; j++)
        {
            System.out.print("A[" + i + "][" +
                             j + "]: ");
            A[i][j] = read.nextInt();
            System.out.println(A[i][j]);
        }
    }

    // Read size of Matrix B from user
    System.out.print("Enter the number of " +
                     "rows of Second Matrix: ");
    row2 = read.nextInt();
    System.out.println(row2);
    System.out.print("Enter the number of " +
                     "columns of Second Matrix: ");
    col2 = read.nextInt();
    System.out.println(col2);
```

Main.java

```java
    // Check if multiplication is Possible
    if (row2 != col1)
    {
        System.out.println("Not Possible");
        return;
    }

    // Multiply the two
    for(i = 0; i < row1; i++)
    {
        for(j = 0; j < col2; j++)
        {
            C[i][j] = 0;
            for(k = 0; k < row2; k++)
                C[i][j] += A[i][k] * B[k][j];
        }
    }

    // Print the result
    System.out.println();
    System.out.println("Resultant Matrix: ");
    printMatrix(C, row1, col2);
}

// Driver code
public static void main(String[] args)
{
    Scanner read = new Scanner(System.in);
    int row1, col1, row2, col2, i, j;
    int A[][] = new int[MAX][MAX];
    int B[][] = new int[MAX][MAX];

    // Read size of Matrix A from user
```

Main.java

```java
    System.out.println(col2);

    // Read the elements of Matrix B from user
    System.out.println("Enter the elements " +
                    "of First Matrix: ");
    for(i = 0; i < row2; i++)
    {
        for(j = 0; j < col2; j++)
        {
            System.out.print("A[" + i + "][" +
                                j + "]: ");
            B[i][j] = read.nextInt();
            System.out.println(B[i][j]);
        }
    }

    // Print the Matrix A
    System.out.println();
    System.out.println("First Matrix: ");
    printMatrix(A, row1, col1);

    // Print the Matrix B
    System.out.println();
    System.out.println("Second Matrix: ");
    printMatrix(B, row2, col2);

    // Find the product of the 2 matrices
    multiplyMatrix(row1, col1, A, row2, col2, B);
}
}

// This code is contributed by Dharanendra L V.
```

Output

```
java -cp /tmp/XZXqXuGuSV GFG
Enter the number of rows of First Matrix: 2
2
Enter the number of columns of First Matrix: 2
2Enter the elements of First Matrix:
A[0][0]: 1 2
1
A[0][1]: 2
A[1][0]: 2 3
2
A[1][1]: 3
Enter the number of rows of Second Matrix: 2
2
Enter the number of columns of Second Matrix: 2
2
Enter the elements of First Matrix:
A[0][0]: 1 2
1
A[0][1]: 2
A[1][0]: 4 5
4
A[1][1]: 5First Matrix:
1 2 2 3

Second Matrix:
1 2
4 5

Resultant Matrix:
9 12
14 19
```

6. ANALYSIS AND DISCUSSION

1) The problem is solved by using Java. The problem was quite hard in the beginning to think out and how to solve this in efficient way. In this program we implement calculations in more efficient way to understand the deeper knowledge of such problems like using files, string, string manipulation, concatenation etc.

7. SUMMARY

1) We have used NetBeans IDE for java
2) We have learned to solve java functionality and some other things from it.

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 04

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 Section: CSE 213 - DA (PC)

Lab Experiment Name(s):

Create a Calculator using Java Program Language.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 09 November, 2022

Submission Date: 29 November, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Graphical Using Interface by Swing

2. OBJECTIVES

The main aim of the swing how to store and use it properly and perform it.

• It is very important for our any project and show it properly as well as how to define size and JFrame and so on.
• Here I actual work frame and input text area, button, Field and so on.

3. PROCEDURE/ ANALYSIS / DESIGN

Algorithm:

1. Step 1: Start
2. Step 2: Create a primary approach
3. Step 3: Create java swing
4. Step 4: Initialize button and textField. Check if the ON / OFF button works and other button works or not.
5. Step 5: Work every button ActionListeners. Implement the arithmetic operations and check hek if they work properly.
6. Step 6: Imported Javax Swing library and others.
7. Step 7: Save this program as jar file.
8. Step 7:End.

Code:

Start Page × calculator.java ×

Source Design History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
package calculator_app;

/**
 *
 * @author shahi
 */
public class calculator extends javax.swing.JFrame {

    /**
     * Creates new form calculator
     */
    double num, ans;
    int calculation;


    public calculator() {
        initComponents();

        jRadioButton1.setEnabled(b: false); //on button disabled
    }


    public void arithmetic_operation()
    {
        switch (calculation)
        {
            case 1:       //add
            ans = num + Double.parseDouble(s: jTextField1.getText());
            jTextField1.setText(s: Double.toString(d: ans));
            break;

            case 2:      //sub
            ans = num - Double.parseDouble(s: jTextField1.getText());
            jTextField1.setText(s: Double.toString(d: ans));
            break;

            case 3:       //mul
            ans = num * Double.parseDouble(s: jTextField1.getText());
            jTextField1.setText(s: Double.toString(d: ans));
```

calculator_app.calculator

Start Page × calculator.java ×

Source Design History

```java
            jTextField1.setText(s: Double.toString(d: ans));
            break;

            case 4:      //div
            ans = num / Double.parseDouble(s: jTextField1.getText());
            jTextField1.setText(s: Double.toString(d: ans));
            break;


        }
    }

    public void enable()
    {
        jTextField1.setEnabled(enabled: true);

        jRadioButton1.setEnabled(b: false);     //on button disable korsi
        jRadioButton2.setEnabled(b: true);      // off button enable korsi


        jButton1.setEnabled(b: true);
        jButton2.setEnabled(b: true);
        jButton3.setEnabled(b: true);
        jButton4.setEnabled(b: true);
        jButton5.setEnabled(b: true);
        jButton6.setEnabled(b: true);
        jButton7.setEnabled(b: true);
        jButton8.setEnabled(b: true);
        jButton9.setEnabled(b: true);
        jButton10.setEnabled(b: true);
        jButton11.setEnabled(b: true);
        jButton12.setEnabled(b: true);
        jButton13.setEnabled(b: true);
        jButton14.setEnabled(b: true);
        jButton15.setEnabled(b: true);
        jButton16.setEnabled(b: true);
        //jButton17.setEnabled(false);
        jButton18.setEnabled(b: true);
        jButton19.setEnabled(b: true);
        jButton20.setEnabled(b: true);
        jButton21.setEnabled(b: true);
        jButton22.setEnabled(b: true);
        jButton23.setEnabled(b: true);
```

calculator_app.calculator

Code:

Start Page × calculator.java ×

Source Design History

```java
            jButton23.setEnabled(b: true);

        }

        public void disable()
        {
            jTextField1.setEnabled(enabled: false);

            jRadioButton1.setEnabled(b: true); // on button enable korsi
            jRadioButton2.setEnabled(b: false); // off button disable

            jButton1.setEnabled(b: false);
            jButton2.setEnabled(b: false);
            jButton3.setEnabled(b: false);
            jButton4.setEnabled(b: false);
            jButton5.setEnabled(b: false);
            jButton6.setEnabled(b: false);
            jButton7.setEnabled(b: false);
            jButton8.setEnabled(b: false);
            jButton9.setEnabled(b: false);
            jButton10.setEnabled(b: false);
            jButton11.setEnabled(b: false);
            jButton12.setEnabled(b: false);
            jButton13.setEnabled(b: false);
            jButton14.setEnabled(b: false);
            jButton15.setEnabled(b: false);
            jButton16.setEnabled(b: false);
            //jButton17.setEnabled(false);
            jButton18.setEnabled(b: false);
            jButton19.setEnabled(b: false);
            jButton20.setEnabled(b: false);
            jButton21.setEnabled(b: false);
            jButton22.setEnabled(b: false);
            jButton23.setEnabled(b: false);


        }

        /**
         * This method is called from within the constructor to initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is always
         * regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
```

calculator_app.calculator

Start Page × calculator.java ×

Source Design History

```java
            jButton21.setEnabled(b: false);
            jButton22.setEnabled(b: false);
            jButton23.setEnabled(b: false);


        }

        /**
         * This method is called from within the constructor to initialize the form.
         * WARNING: Do NOT modify this code. The content of this method is always
         * regenerated by the Form Editor.
         */
        @SuppressWarnings("unchecked")
        Generated Code

        private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {

            enable(); // enable method kall korsi
        }

        private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(s: "");
        }

        private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
            num = Double.parseDouble(s: jTextField1.getText());
            calculation = 1;
            jTextField1.setText(s: "");
            jLabel1.setText(num+"+");

        }

        private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
          jTextField1.setText(jTextField1.getText() +"7");
        }

        private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
          jTextField1.setText(jTextField1.getText() +"8");
        }

        private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
            jTextField1.setText(jTextField1.getText() +"9");
        }
```

calculator_app.calculator

Code:

Start Page × calculator.java ×

Source Design History

```
    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
      jTextField1.setText(jTextField1.getText() +"8");
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
       jTextField1.setText(jTextField1.getText() +"9");
    }

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {

        num = Double.parseDouble( s: jTextField1.getText());
        calculation = 2;
        jTextField1.setText( t: "");
        jLabel1.setText(num+"-");
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton11ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton12ActionPerformed(java.awt.event.ActionEvent evt) {
        num = Double.parseDouble( s: jTextField1.getText());
        calculation = 3;
        jTextField1.setText( t: "");
        jLabel1.setText(num+"*");
    }

    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
       jTextField1.setText(jTextField1.getText() +"4");
    }

    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
```

calculator_app.calculator

Start Page × calculator.java ×

Source Design History

```
        jTextField1.setText( t: "");
        jLabel1.setText(num+"*");
    }

    private void jButton13ActionPerformed(java.awt.event.ActionEvent evt) {
       jTextField1.setText(jTextField1.getText() +"4");
    }

    private void jButton14ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.setText(jTextField1.getText() +"5");
    }

    private void jButton15ActionPerformed(java.awt.event.ActionEvent evt) {
       jTextField1.setText(jTextField1.getText() +"6");
    }

    private void jButton16ActionPerformed(java.awt.event.ActionEvent evt) {
         num = Double.parseDouble( s: jTextField1.getText());
        calculation = 4;
        jTextField1.setText( t: "");
        jLabel1.setText(num+"/");
    }

    private void jButton18ActionPerformed(java.awt.event.ActionEvent evt) {
      jTextField1.setText(jTextField1.getText() +"0");
    }

    private void jButton19ActionPerformed(java.awt.event.ActionEvent evt) {
      jTextField1.setText(jTextField1.getText() +".");
    }

    private void jButton20ActionPerformed(java.awt.event.ActionEvent evt) {
        arithmetic_operation();
        jLabel1.setText( text: "");

        // TODO add your handling code here:
    }

    private void jButton21ActionPerformed(java.awt.event.ActionEvent evt) {
       jTextField1.setText(jTextField1.getText()+ "1");
    }

    private void jButton22ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextField1.setText(jTextField1.getText() +"2");
```

calculator_app.calculator

Code:

Start Page × calculator.java ×

Source Design History

```java
        jTextfield1.setText(jTextfield1.getText()+ "1");
    }

    private void jButton22ActionPerformed(java.awt.event.ActionEvent evt) {
    jTextfield1.setText(jTextfield1.getText() +"2");

    }

    private void jButton23ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextfield1.setText(jTextfield1.getText() +"3");
    }

    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
   disable(); //disbale method kall

// TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        int length = jTextfield1.getText().length();
        int number = jTextfield1.getText(). length();
        String store;

        if (length>0)
        {
            StringBuilder back = new StringBuilder(str:jTextfield1.getText());
            back.deleteCharAt(index:number);
            store=back.toString();
            jTextfield1.setText(t:store);

        }



    }

    private void jTextfield1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
```

calculator_app.calculator

Start Page × calculator.java ×

Source Design History

```java
    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new calculator().setVisible(b:true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.ButtonGroup buttonGroup1;
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton10;
    private javax.swing.JButton jButton11;
    private javax.swing.JButton jButton12;
    private javax.swing.JButton jButton13;
    private javax.swing.JButton jButton14;
    private javax.swing.JButton jButton15;
    private javax.swing.JButton jButton16;
    private javax.swing.JButton jButton18;
    private javax.swing.JButton jButton19;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton20;
    private javax.swing.JButton jButton21;
    private javax.swing.JButton jButton22;
    private javax.swing.JButton jButton23;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JButton jButton9;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JRadioButton jRadioButton1;
    private javax.swing.JRadioButton jRadioButton2;
```

calculator_app.calculator

Code:

Start Page × calculator.java ×

Source Design History

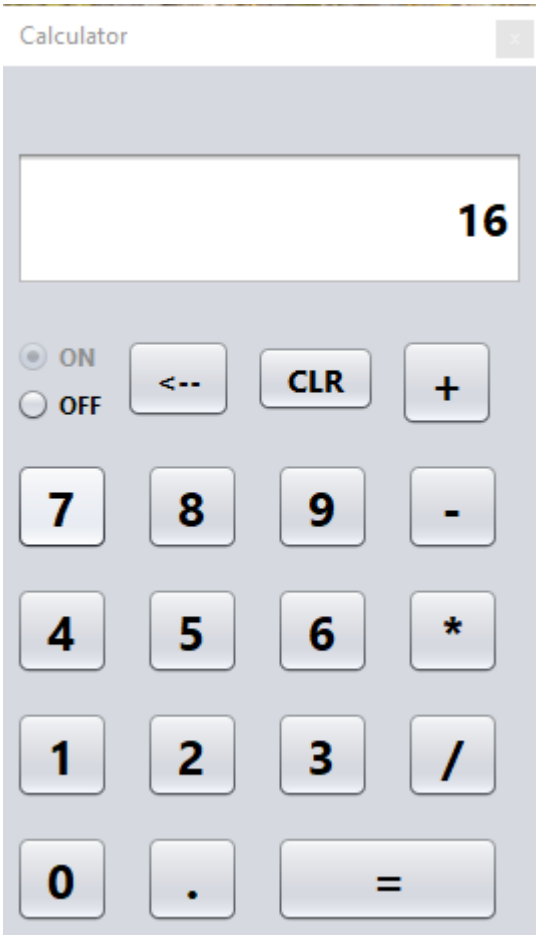
```
650         private javax.swing.JButton jButton10;
651         private javax.swing.JButton jButton11;
652         private javax.swing.JButton jButton12;
653         private javax.swing.JButton jButton13;
654         private javax.swing.JButton jButton14;
655         private javax.swing.JButton jButton15;
656         private javax.swing.JButton jButton16;
657         private javax.swing.JButton jButton18;
658         private javax.swing.JButton jButton19;
659         private javax.swing.JButton jButton2;
660         private javax.swing.JButton jButton20;
661         private javax.swing.JButton jButton21;
662         private javax.swing.JButton jButton22;
663         private javax.swing.JButton jButton23;
664         private javax.swing.JButton jButton3;
665         private javax.swing.JButton jButton4;
666         private javax.swing.JButton jButton5;
667         private javax.swing.JButton jButton6;
668         private javax.swing.JButton jButton7;
669         private javax.swing.JButton jButton8;
670         private javax.swing.JButton jButton9;
671         private javax.swing.JLabel jLabel1;
672         private javax.swing.JLabel jLabel2;
673         private javax.swing.JRadioButton jRadioButton1;
674         private javax.swing.JRadioButton jRadioButton2;
675         private javax.swing.JTextField jTextField1;
676         // End of variables declaration
677     }
678
```

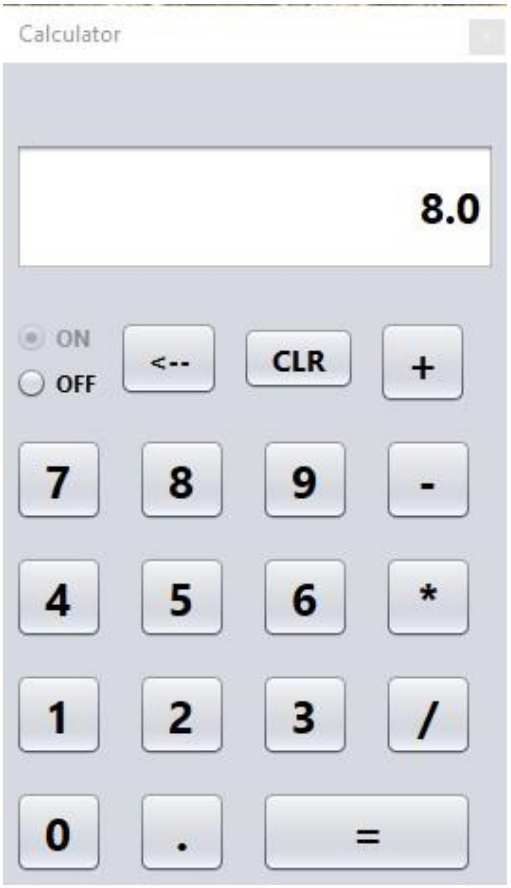
calculator_app.calculator

Output:

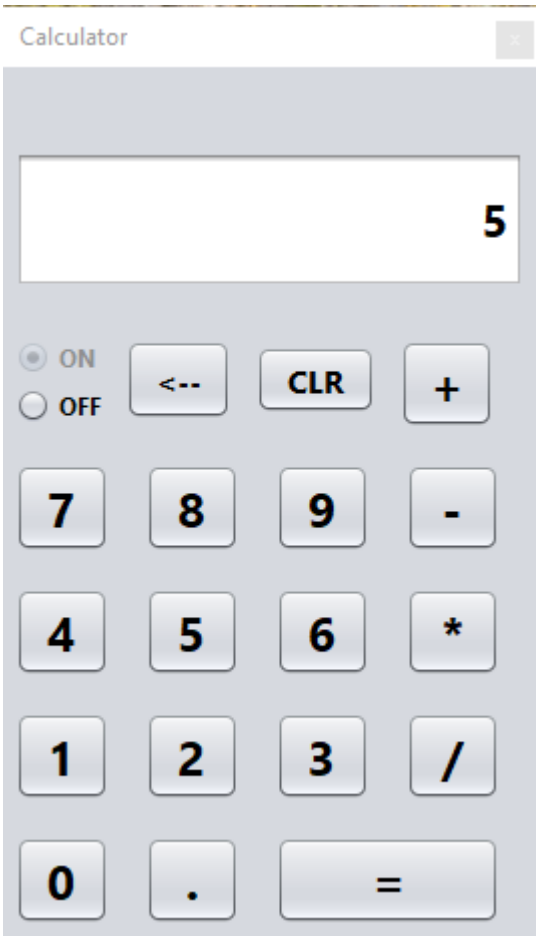

Calculator
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=

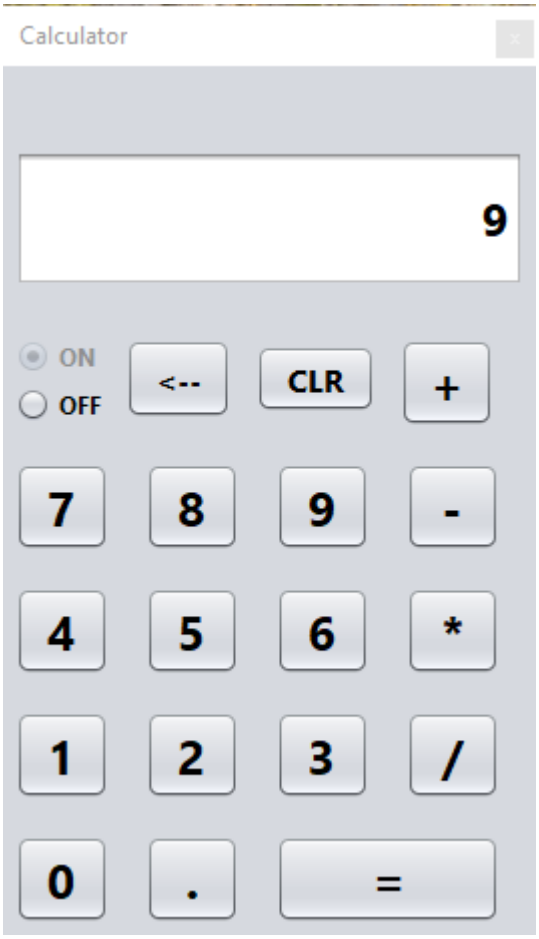

Calculator
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=

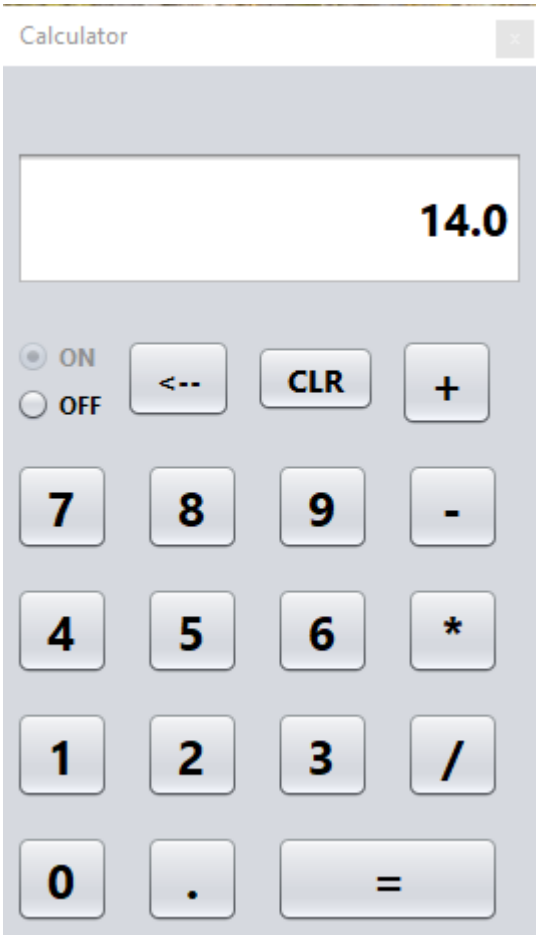

Calculator
16
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=


Calculator
8.0
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=

Output:


Calculator
5
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=


Calculator
9
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=


Calculator
14.0
ON
OFF
<--
CLR
+
7
8
9
-
4
5
6
*
1
2
3
/
0
.
=

Output:

Test	Input	Expected output	Original output	Result
Addition	5+9	14	14.0	Pass
Subtraction	10-5	5	5.0	Pass
Multiplication	10*6	60	60.0	Pass
Division	16/2	8	8.0	Pass

6. ANALYSIS AND DISCUSSION

1). We could not show how to define different set bounds.
2) I could not show the display while a number is stored
3) Implemented the basic arithmetic functions only.

7. SUMMARY

1) We have used NetBeans IDE for java
2) We have learned to solve java functionality and some other things from it.
3) We have learned the usage of Swing and saved it as a JAR file.

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 06

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 **Section: CSE 213 - DA (PC)**

Lab Experiment Name(s):

Polymorphism

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 28 November, 2022

Submission Date: 04 December, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Polymorphism

2. OBJECTIVES

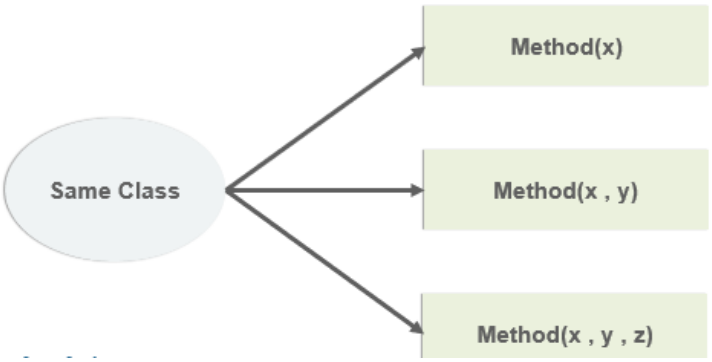
- Understanding Polymorphism in Abstract classes
- Method overloading, Method overriding

3. DEFINITION & DESCRIPTION

What is Method Overloading in Java?

Method overloading allows the method to have the same name which differs on the basis of arguments or the argument types. It can be related to compile-time polymorphism. Following are a few pointers that we have to keep in mind while overloading methods in Java.

- We cannot overload a return type.
- Although we can overload static methods, the arguments or input parameters have to be different.
- We cannot overload two methods if they only differ by a static keyword.
- Like other static methods, the main() method can also be overloaded.



3. DEFINITION & DESCRIPTION

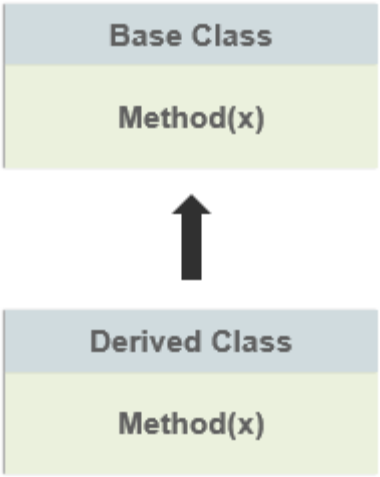
What Is Method Overriding in Java?

Inheritance in java involves a relationship between parent and child classes. Whenever both the classes contain methods with the same name and arguments or parameters it is certain that one of the methods will override the other method during execution. The method that will be executed depends on the object.

If the child class object calls the method, the child class method will override the parent class method. Otherwise, if the parent class object calls the method, the parent class method will be executed.

Rules For Method Overriding

- ❑ The access modifier can only allow more access for the overridden method.
- ❑ A final method does not support method overriding.
- ❑ A static method cannot be overridden.
- ❑ Private methods cannot be overridden.
- ❑ The return type of the overriding method must be the same.
- ❑ We can call the parent class method in the overriding method using the super keyword.
- ❑ A constructor cannot be overridden because a child class and a parent class cannot have the constructor with the same name.



4. IMPLEMENTATION & TEST RESULT

Example of Method Overloading

Main.java

```java
import java.io.*;

class MethodOverloadingEx {

    static int add(int a, int b)
    {
    return a + b;
    }

    static int add(int a, int b, int c)
    {
        return a + b + c;
    }

    public static void main(String args[])
    {
        System.out.println("add() with 2 parameters");
        System.out.println(add(4, 6));

        System.out.println("add() with 3 parameters");
        System.out.println(add(4, 6, 7));
    }
}
```

Output

```
java -cp /tmp/jvTeOTjx65 MethodOverloadingEx
add() with 2 parameters
10
add() with 3 parameters
17
```

4. IMPLEMENTATION & TEST RESULT

Example of Method Overriding

Main.java

```java
import java.io.*;
class Animal {
    void eat()
    {
        System.out.println("eat() method of base class");
        System.out.println("eating.");
    }
}
class Dog extends Animal {
    void eat()
    {
        System.out.println("eat() method of derived class");
        System.out.println("Dog is eating.");
    }
}
class MethodOverridingEx {
    public static void main(String args[])
    {
        Dog d1 = new Dog();
        Animal a1 = new Animal();

        d1.eat();
        a1.eat();

        Animal animal = new Dog();

        animal.eat();
    }
}
```

Output

```
java -cp /tmp/jvTeOTjx65 MethodOverridingEx
eat() method of derived class
Dog is eating.
eat() method of base class
eating.
eat() method of derived class
Dog is eating.
```

5. ANALYSIS AND DISCUSSION

1) Here, we can see that a method eat() has overridden in the derived class name **Dog** that is already provided by the base class name **Animal**. When we create the instance of class Dog and call the eat() method, we see that only derived class eat() method run instead of base class method eat(), and When we create the instance of class Animal and call the eat() method, we see that only base class eat() method run instead of derived class method eat().

6. SUMMARY

Overriding occurs when the method signature is the same in the superclass and the child class. Overloading occurs when two or more methods in the same class have the same name but different parameters..

Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Faculty of Sciences and Engineering
Fall 2022, B.Sc. in CSE (DAY)

LAB REPORT NO # 07

Course Title: Object Oriented Programming (JAVA)
Course Code: CSE 202 **Section: CSE 213 - DA (PC)**

Lab Experiment Name(s):

Interface

- Create an interface isEmergency with only one method - soundSiren which takes no arguments and returns no value.
- Write a class FireEmergency that implements the IsEmergency interface. The soundSiren method should print "Siren Sounded".
- Write a class SmokeAlarm that does not implement any interface. The class has an empty body.
- Create an array of Object class, myArray in the main method.
- Construct 2 SmokeAlarm object and add it to the array myArray in the main method.
- Construct 2 FireEmergency object and add it to the array myArray in the main method.
- In the main method, write a for loop, to print which array elements are instances of classes that implement the IsEmergency interface and if so, call the soundSiren method.

Student Details

Name	ID
Md. Shahidul Islam Prodhan	213902017

Lab Date: 05 December, 2022

Submission Date: 12 December, 2022

Course Teacher's Name: Dr. Muhammad Aminur Rahaman, Associate Professor

[For Teacher's use only: Don't write anything inside this box]

Lab Report Status

Marks:	**Signature:**
Comments:	**Date:**

1. TITLE OF THE LAB EXPERIMENT

Java Interface

2. OBJECTIVES

1. Using a class that implements several interfaces, we can comprehend multiple inheritance.
2. Through the use of an abstract type, an interface in java describes how a class should behave.
3. Specifying methods that a class or classes must implement.
4. Then to improve the readability and comprehension of the code.

3. ALGORITHM:

Steps	Procedures / Works
Step 1	start
Step 2	Create an interface IsEmergency with an abstract technique soundSiren.
Step 3	Create a class FireEmergency that implements IsEmergency .
Step 4	Override the soundSiren method inside the FireEmergency elegance.
Step 5	Create every other magnificence SmokeAlarm with an empty frame.
Step 6	Create a primary class and put into effect the primary technique.
Step 7	Inside the principal technique create an array of item magnificence and create 2 objects of SmokeAlarm and 2 objects of FireEmergency .
Step 8	Write a for loop, to print which array elements are instances of training that implement the IsEmergency interface and if so, name the soundSiren method.
Step 9	End

4. IMPLEMENTATION & TEST RESULT

Example of Implementing Interface

CODE

Start Page × Jframe.java × Students.java × BioData.java × Test.java × FireEmergency.java × **FireEmergency.java** ×

Source History

```java
/**
 * @author shahidul
 */
interface isEmergency{
void soundSiren();
}
class FireEmergency implements isEmergency{
@override
public void soundSiren(){
    System.out.println( x:"Siren Sounded");
}
        }

class SmokeAlarm{
}
public class FireEmergency {

    public static void main(String[] args) {

        SmokeAlarm p1 = new SmokeAlarm();
        SmokeAlarm p2 = new SmokeAlarm();
        FireEmergency p3 = new FireEmergency();
        FireEmergency p4 = new FireEmergency();
        FireEmergency[] myArray = {p3, p4};

        for (FireEmergency obj : myArray){
            obj.soundSiren();
        }
    }
}
```

OUTPUT

I could not successfully show expected output results in my Netbeans IDE.

4. IMPLEMENTATION & TEST RESULT

Example of Implementing Interface

OUTPUT

I could not successfully show expected output results in my Netbeans IDE.
There was error in my program.

Start Page | FireEmergency.java

Source | History

```
6   }
    class FireEmergency implements isEmergency{
    @override
    public void soundSiren(){
10      System.out.println( x:"Siren Sounded");
11  }
12          }
13
    class SmokeAlarm{
15  }
    public class FireEmergency {
17
18      public static void main(String[] args) {
19
20          SmokeAlarm p1 = new SmokeAlarm();
21          SmokeAlarm p2 = new SmokeAlarm();
22          FireEmergency p3 = new FireEmergency();
23          FireEmergency p4 = new FireEmergency();
```

FireEmergency

Output - Run (FireEmergency)

```
cd C:\Users\shahi\Documents\NetBeansProjects\FireEmergency; "JAVA_HOME=C:\\Program Files\\Java\\jdk-19" cmd /c "\"C:\\Prog
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with
Scanning for projects...

--------------------< com.mycompany:FireEmergency >---------------------
Building FireEmergency 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ FireEmergency ---
Error: Could not find or load main class com.mycompany.fireemergency.FireEmergency
Caused by: java.lang.ClassNotFoundException: com.mycompany.fireemergency.FireEmergency
Command execution failed.
org.apache.commons.exec.ExecuteException: Process exited with an error: 1 (Exit value: 1)
    at org.apache.commons.exec.DefaultExecutor.executeInternal (DefaultExecutor.java:404)
    at org.apache.commons.exec.DefaultExecutor.execute (DefaultExecutor.java:166)
    at org.codehaus.mojo.exec.ExecMojo.executeCommandLine (ExecMojo.java:982)
    at org.codehaus.mojo.exec.ExecMojo.executeCommandLine (ExecMojo.java:929)
    at org.codehaus.mojo.exec.ExecMojo.execute (ExecMojo.java:457)
    at org.apache.maven.plugin.DefaultBuildPluginManager.executeMojo (DefaultBuildPluginManager.java:137)
    at org.apache.maven.lifecycle.internal.MojoExecutor.doExecute2 (MojoExecutor.java:370)
    at org.apache.maven.lifecycle.internal.MojoExecutor.doExecute (MojoExecutor.java:351)
    at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:215)
    at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:171)
    at org.apache.maven.lifecycle.internal.MojoExecutor.execute (MojoExecutor.java:163)
    at org.apache.maven.lifecycle.internal.LifecycleModuleBuilder.buildProject (LifecycleModuleBuilder.java:117)
    at org.apache.maven.lifecycle.internal.LifecycleModuleBuilder.buildProject (LifecycleModuleBuilder.java:81)
```

5. ANALYSIS AND DISCUSSION

1) Using a class that implements various interfaces, the challenge illustrates multiple inheritance.
2) We learnt, how to construct an item and add it to the array while doing interface code.