



**KK14203 INDIVIDUAL PROJECT II**  
**STUDENT NAME : MCLAREN W. YUSOF**  
**STUDENT NUMBER : BI19110151**

## INTRODUCTION

This individual project was assigned to the students by Mr Mohd Shamrie Sainin, a faculty member and a lecturer of Faculty of Computing and Informatics of UMS as means of students' assessment.

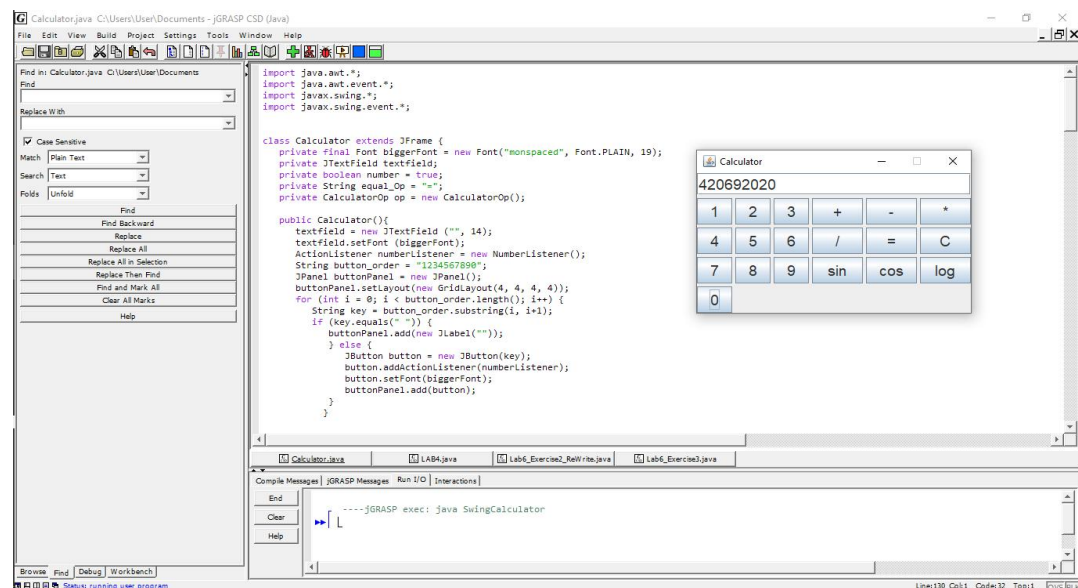
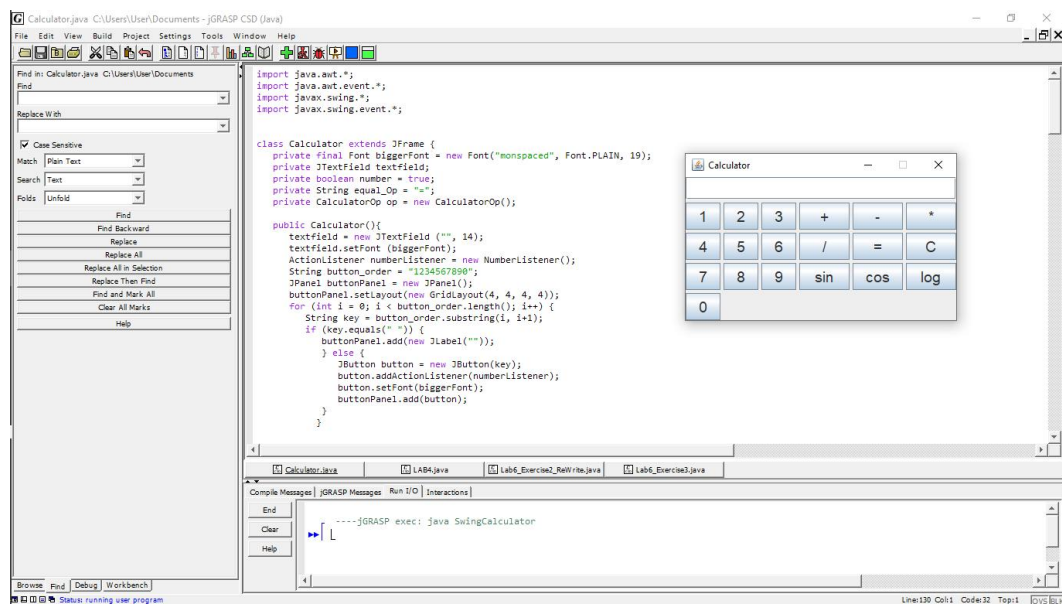
My purpose of writing this introduction is to explain about how this project works and the implementations of the knowledge of object-oriented programming throughout the java code created. The classes written as code in java of this program are *class* Calculator, *class* OperatorListener, *class* NumberListener, *public class* CalculatorOp, and *class* SwingCalculator. Objects are not implemented. Encapsulation as well as inheritance is implemented in the program, thus being able to reduce disturbance in the code and minimize errors. Creation of GUI is also made possible. Lastly, methods are also commonly implemented throughout this program as such example in *class* NumberListener *implements* Action Listener where the code is written as :

```
public void actionPerformed(ActionEvent event) {  
    String digit = event.getActionCommand();  
    if (number) {  
        textfield.setText(digit);  
        number = false;  
    } else {  
        textfield.setText(textfield.getText() + digit);  
    }  
}
```

## READ AND WRITE IMPLEMENTATION

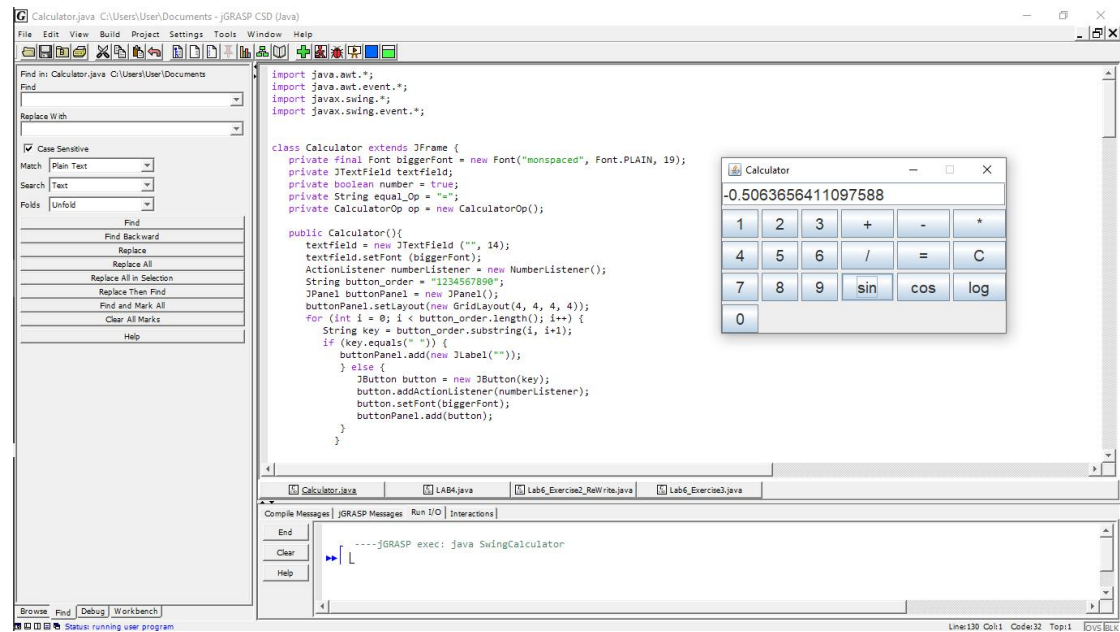
For this particular implementation, the examples can be referred below (using pictures and simple explanatory statements).

- The simple calculator can be seen when you run the java program.
- It shows digits 0 to 9, arithmetic operation symbols, logarithm, trigonometric symbols of sin and cos and the clear all ( C ) button.
- The calculator starts off with a blank top above the number buttons which fills numbers in accordance to the user clicking the specific buttons by using the mouse.
- Operations in numbers below 0 are not suitable as this is by no means a scientific calculator.
- The blank top bar can be inputted by using the keyboard but is highly not recommended.

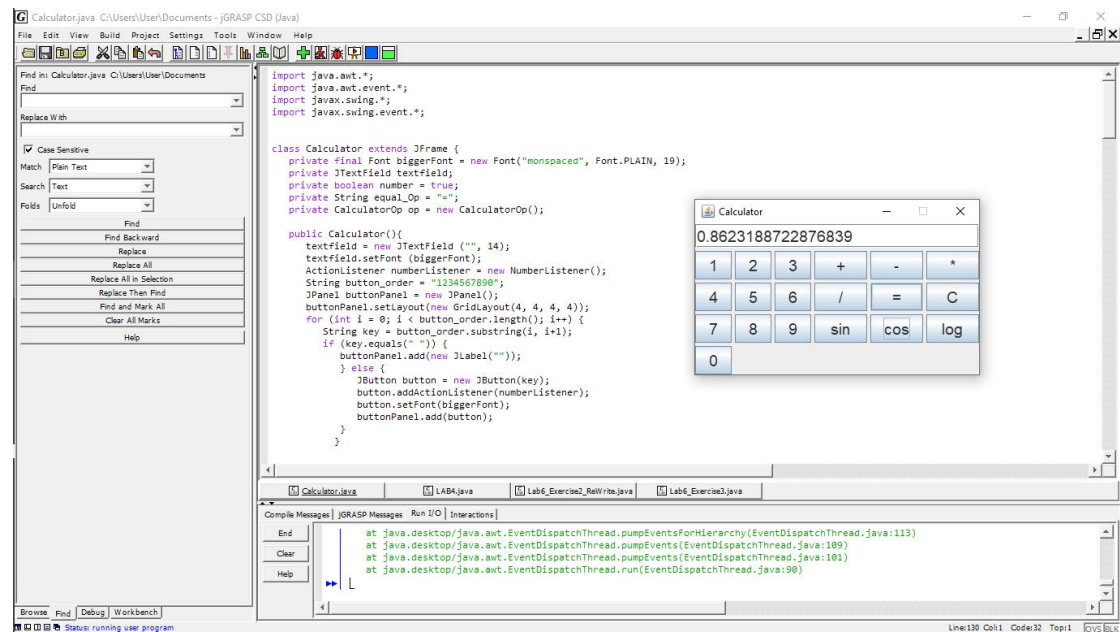


- When the user clicks the number buttons for example, the top bar is filled as shown above.
- In order to allow arithmetic operations of simple numbers, the user has to click on the wanted numbers, followed by the arithmetic operation symbols, and then numbers once again, and finally the equality sign.

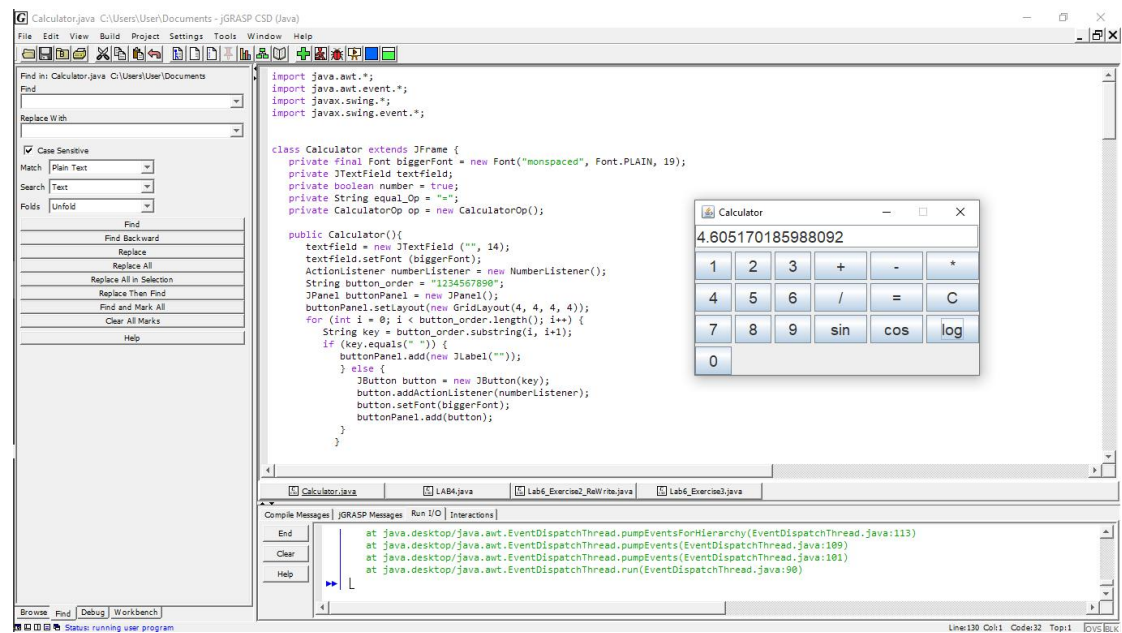
- For your information, the calculator operates on numbers as long as the user does not click on the clear all ( C ) button, thus making mathematics longer.



- The above picture is shown when the user inputs a specific number followed by the 'sin' button.
- Operations in trigonometry or logarithm are not recommended when using this calculator. Once again, it is not a scientific calculator.



- The above picture shows the numbers on the top bar when the user clicks on specific number buttons followed by the 'cos' button.



- The figure above shows the top bar of the calculator when the user clicks on specific number buttons followed by the 'log' button.

## SOURCE CODE (JAVA)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

class Calculator extends JFrame {
    private final Font biggerFont = new Font("monospaced", Font.PLAIN, 19);
    private JTextField textfield;
    private boolean number = true;
    private String equal_Op = "=";
    private CalculatorOp op = new CalculatorOp();

    public Calculator(){
        textfield = new JTextField("", 14);
        textfield.setFont(biggerFont);
        ActionListener numberListener = new NumberListener();
        String button_order = "1234567890";
        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(4, 4, 4, 4));
        for (int i = 0; i < button_order.length(); i++) {
            String key = button_order.substring(i, i+1);
            if (key.equals(" ")) {
                buttonPanel.add(new JLabel(""));
            } else {
                JButton button = new JButton(key);
                button.addActionListener(numberListener);
                button.setFont(biggerFont);
                buttonPanel.add(button);
            }
        }
        ActionListener operatorListener = new OperatorListener();
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(4, 4, 4, 4));
        String[] opOrder = {"+", "-", "*", "/", "=", "C", "sin", "cos", "log"};
        for (int i = 0; i < opOrder.length; i++) {
            JButton button = new JButton(opOrder[i]);
            button.addActionListener(operatorListener);
            button.setFont(biggerFont);
            panel.add(button);
        }
        JPanel pan = new JPanel();
        pan.setLayout(new BorderLayout(4, 4));
        pan.add(textfield, BorderLayout.NORTH);
        pan.add(buttonPanel, BorderLayout.CENTER);
        pan.add(panel, BorderLayout.EAST);
        this.setContentPane(pan);
        this.pack();
        this.setTitle("Calculator");
        this.setResizable(false);
    }

    private void action() {
        number = true;
        textfield.setText("");
    }
}
```

```

        equal_Op = "=";
        op.setTotal("");
    }
    class OperatorListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            String displayText = textfield.getText();
            if (e.getActionCommand().equals("sin"))
            {
                textfield.setText("" + Math.sin(Double.valueOf(displayText).doubleValue()));

            }else
            if (e.getActionCommand().equals("cos"))
            {
                textfield.setText("" + Math.cos(Double.valueOf(displayText).doubleValue()));

            }
            else
            if (e.getActionCommand().equals("log"))
            {
                textfield.setText("" + Math.log(Double.valueOf(displayText).doubleValue()));

            }
            else if (e.getActionCommand().equals("C"))
            {
                textfield.setText("");
            }

            else
            {
                if (number)
                {
                    action();
                    textfield.setText("");
                }
                else
                {
                    number = true;
                    if (equal_Op.equals("="))
                    {
                        op.setTotal(displayText);
                    }else
                    if (equal_Op.equals("+"))
                    {
                        op.add(displayText);
                    }
                    else if (equal_Op.equals("-"))
                    {
                        op.subtract(displayText);
                    }
                    else if (equal_Op.equals("*"))
                    {
                        op.multiply(displayText);
                    }
                    else if (equal_Op.equals("/"))
                    {

```

```

        op.divide(displayText);
    }

    textfield.setText("" + op.getTotalString());
    equal_Op = e.getActionCommand();
}
}

class NumberListener implements ActionListener {
    public void actionPerformed(ActionEvent event) {
        String digit = event.getActionCommand();
        if (number) {
            textfield.setText(digit);
            number = false;
        } else {
            textfield.setText(textfield.getText() + digit);
        }
    }
}

public class CalculatorOp {
    private int total;
    public CalculatorOp() {
        total = 0;
    }
    public String getTotalString() {
        return ""+total;
    }
    public void setTotal(String n) {
        total = convertToNumber(n);
    }
    public void add(String n) {
        total += convertToNumber(n);
    }
    public void subtract(String n) {
        total -= convertToNumber(n);
    }
    public void multiply(String n) {
        total *= convertToNumber(n);
    }
    public void divide(String n) {
        total /= convertToNumber(n);
    }
    private int convertToNumber(String n) {
        return Integer.parseInt(n);
    }
}

}

class SwingCalculator {
    public static void main(String[] args) {
        JFrame frame = new Calculator();
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}

```