

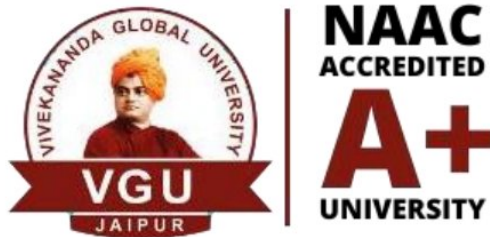
Calculator Application With Basic Airthmethic Operations

A6 PROJECT REPORT
of
Object Oriented Programming Using Java
PGCSA111

*Submitted in partial fulfilment of the requirement for the award of Degree
of*

MASTER OF COMPUTER APPLICATIONS

Submitted to



Submitted to:
Mr. Narayan Vyas
Computer Science & Application

Submitted by:
Abhinav Kumar (24CSA3BC110)
Akshita Sharma (24CSA3BC001)
Abhay Jangir (24CSA3BC053)

Department of Computer Science & Application
Vivekananda Global University, Jaipur
Year- 2025

Introduction of the Project

If you're looking to dive into the basics of Java and want to create a practical application, then building a simple calculator is a fantastic starting point. Calculators are fundamental tools used in many applications, and building one in Java can be a great way to improve your programming skills. As a Java expert, let me guide you through the process of developing a simple calculator using Java.

In this guide, we'll walk you through the step-by-step process of developing a simple calculator using Java. You'll learn how to set up your Java development environment, create a Java class, get user input, perform calculations, display results, add error handling, and test your program. Whether you're a beginner or an experienced Java developer, this guide will provide you with the knowledge and tools to build a functional calculator that you can be proud of.

So, let's dive in and unlock the world of Java programming as we start on this exciting journey to develop a simple calculator that showcases the power and versatility of Java!

Objectives

1. To implement the right mathematics to solve simple problems of addition, subtraction, multiplication, and division.
2. To create a graphic user interface (GUI) involving some graphics, along with the fundamental calculator logic, to give users a good interface to work with.

Requirements

Before diving into the process of developing a simple calculator using Java, there are a few prerequisites that you need to have in place. These include:

1. Java Development Kit (JDK): You'll need to have the JDK installed on your computer. The JDK includes the Java compiler and runtime environment necessary for developing and running Java programs. You can [download the JDK](#) from the official Java website and follow the installation instructions.

2. Integrated Development Environment (IDE): You'll need an IDE to write, compile, and run your Java code. There are several popular IDEs for Java, such as Eclipse, [IntelliJ IDEA](#), and NetBeans. Choose an IDE that you are comfortable with, or try out a few to see which one suits your preferences.

3. Basic understanding of Java syntax: It's essential to have a basic understanding of Java syntax, including data types, variables, operators, control flow statements (such as ifelse and loops), and basic object-oriented programming concepts (such as classes and methods).

4. Understanding of arithmetic operations: You should have a solid understanding of basic arithmetic operations such as addition, subtraction, multiplication, and division, as you'll be using these operations to perform calculations in your calculator.

5. Swing module to implement the Graphic User Interface: The best part about using Swing to create components like tables, scroll panes, buttons, and the like is that the framework is platform-independent.

Source Code

```
// Java program to create a simple calculator // with basic +,  
-, /, * using java swing elements import java.awt.event.*;  
import javax.swing.*; import java.awt.*; class calculator  
extends JFrame implements ActionListener {  
// Creating a frame  
static JFrame f; //  
Creating a textfield  
static JTextField l;  
// storing operator and operands  
String s0, s1, s2;  
// The default constructor calculator()  
{  
s0 = s1 = s2 = "";  
}  
// Main function public static void  
main(String args[])  
{  
// create a frame f = new  
JFrame("calculator"); try {/  
Setting the look &  
feel UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());  
}  
catch (Exception e) {  
System.err.println(e.getMessage());  
}
```

```

// creating an object of class
calculator c = new calculator();

// creating a textfield l = new
JTextField(16);

// setting the textfield to non-editable
l.setEditable(false);

// creating buttons for the numbers and operators
JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq, beq1;

// number buttons b0 = new
JButton("0"); b1 = new
JButton("1"); b2 = new
JButton("2"); b3 = new
JButton("3"); b4 = new
JButton("4"); b5 = new
JButton("5"); b6 = new
JButton("6"); b7 = new
JButton("7"); b8 = new
JButton("8"); b9 = new
JButton("9"); // creating the
equals to button beq1 = new
JButton("=");

// creating buttons for the operator
ba = new JButton("+"); bs = new
JButton("-"); bd = new JButton("/");
bm = new JButton("*"); beq = new
JButton("C"); // creating decimal

```

```
"." button be = new JButton("."); //
```

creating one panel

```
JPanel p = new JPanel(); //
```

adding the action listeners

```
bm.addActionListener(c);
```

```
bd.addActionListener(c);
```

```
bs.addActionListener(c);
```

```
ba.addActionListener(c);
```

```
b9.addActionListener(c);
```

```
b8.addActionListener(c);
```

```
b7.addActionListener(c);
```

```
b6.addActionListener(c);
```

```
b5.addActionListener(c);
```

```
b4.addActionListener(c);
```

```
b3.addActionListener(c);
```

```
b2.addActionListener(c);
```

```
b1.addActionListener(c);
```

```
b0.addActionListener(c);
```

```
be.addActionListener(c);
```

```
beq.addActionListener(c);
```

```
beq1.addActionListener(c);
```

```
// adding different elements
```

to the panel

```
p.add(l);
```

```
p.add(ba);
```

```
p.add(b1);
```

```
p.add(b2);
```

```
p.add(b3);
```

```
p.add(bs);
```

```
p.add(b4);  
p.add(b5);  
p.add(b6);  
p.add(bm);  
p.add(b7);  
p.add(b8);  
p.add(b9);  
p.add(bd);  
p.add(be);  
p.add(b0);  
p.add(beq);  
p.add(beq1);
```

```
// setting the Background of the panel
```

```
p.setBackground(Color.black);
```

```
// adding the panel to the frame
```

```
f.add(p);
```

```
f.setSize(200, 220);
```

```
f.show();
```

```
}
```

```
public void actionPerformed(ActionEvent e)
```

```
{
```

```
String s = e.getActionCommand(); // if the value entered is a
```

```
number if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') ||
```

```
s.charAt(0) == '.') { // if operand is present then add to second  
number
```

```

if (!s1.equals(""))
    s2 = s2 + s; else
    s0 = s0 + s;
// setting the value of the text
l.setText(s0 + s1 + s2);
}
else if (s.charAt(0) == 'C') {
// clearing the one letter
s0 = s1 = s2 = ""; // set
the value of text
l.setText(s0 + s1 + s2);
}
else if (s.charAt(0) == '=') {
double te;
// store the value in 1st if
(s1.equals("+"))
te = (Double.parseDouble(s0) + Double.parseDouble(s2));
else if (s1.equals("-"))
te = (Double.parseDouble(s0) - Double.parseDouble(s2));
else if (s1.equals("/"))
te = (Double.parseDouble(s0) / Double.parseDouble(s2));
else
te = (Double.parseDouble(s0) * Double.parseDouble(s2));
// set the value of text
l.setText(s0 + s1 + s2 + "=" + te);
// converting it to string s0 =
Double.toString(te); s1 = s2 =
"",
}

```

```

else {
// if there was no operand if
(s1.equals("") || s2.equals(""))
s1 = s; // else evaluate
else {
double te; // store the
value in 1st if
(s1.equals("+"))
te = (Double.parseDouble(s0) + Double.parseDouble(s2));
else if (s1.equals("-"))
te = (Double.parseDouble(s0) - Double.parseDouble(s2));
else if (s1.equals("/"))
te = (Double.parseDouble(s0) / Double.parseDouble(s2));
else te = (Double.parseDouble(s0) *
Double.parseDouble(s2));
// convert it to string s0 =
Double.toString(te); // place the
operator s1 = s; // make the
operand blank
s2 = "";
}
// set the value of text
l.setText(s0 + s1 + s2);
}
}
}

```

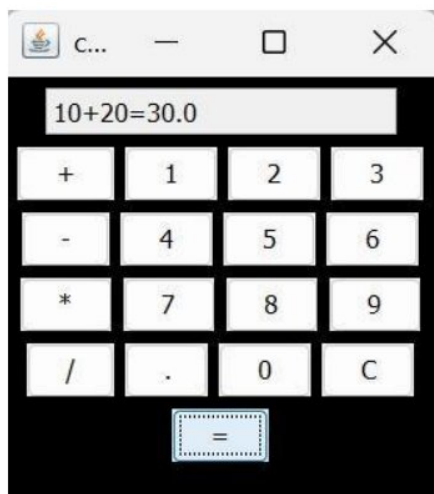
Explanation of the Code

Since we will be using Swing to create the GUI of the application, you can imagine that will be the major chunk of the program as the mathematics is pretty simple since we will restrict this basic calculator to +, -, *, and / operations.

1. For the GUI, we will start by creating buttons for all the digits from 0-9. We will also create buttons for the operands such as +, -, *, and /.
2. The application will have a button to clear the values from the “display” of the calculator application.
3. We will also use a text field that will be set to uneditable. This text field will become the display of the calculator.
4. Once all the buttons have been created and laid out in the JFrame, the buttons will be functionalized using the [addActionListener](#) (ActionListener d) function.
5. Whenever a button containing any digit is pressed, we will go to an **“IF statement”** that will store it in a variable. If an operator button is pressed, it will be stored as a string again.
6. If the “=” button is pressed, the program will check for what kind of operation and perform the operation. The output will be converted back to string and displayed as output.

Output

Addition



Multiplication

