CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTE OF COMPUTING

PROJECT REPORT
ON
Simple Calculator Application

Program Name: BCA

Subject Name/Code: Linux Administration Lab(23CAP-305)

Submitted by:	**Submitted to:**
Name: Sanjana Chauhan	**Name:** Mr. Rajat Kapoor
UID: 23BCA10150	**Designation:** AP
Section: BCA – 2 “B”	

Simple Calculator Application

1. Title of the Project

Simple Calculator using C++ in Linux Environment

2. Objective of the Project

The main objective of this project is to design and implement a **simple calculator** using the **C++ programming language** that can perform basic arithmetic operations such as **addition, subtraction, multiplication, and division**.
The project also demonstrates the use of fundamental programming concepts like **input/output, switch-case statements, operators, and conditional statements**.

3. Software and Hardware Requirements

Software Requirements

- **Operating System:** Ubuntu Linux (running in VirtualBox)
- **Compiler:** GNU C++ Compiler (g++)
- **Editor:** nano or any text editor
- **Virtualization Platform:** Oracle VM VirtualBox

Hardware Requirements

- Processor: Intel or AMD Dual Core (or higher)
- RAM: Minimum 2 GB
- Hard Disk: Minimum 10 GB free space
- Display: Standard VGA or higher

4. Tools Used

- **VirtualBox** – To run a Linux virtual environment on a host system.
- **Ubuntu Terminal** – For compiling and executing the C++ program.
- **g++ Compiler** – To compile the C++ source code into an executable program.
- **nano Editor** – To write and edit source code within the terminal.

5. Methodology

Step 1: Program Design

The calculator accepts two numerical inputs and one operator from the user.
It then performs the operation based on the entered operator using a **switch-case** structure.

Step 2: Program Flow

1. Start the program.
2. Take the first number as input.
3. Take an operator as input (+, -, *, /).
4. Take the second number as input.
5. Perform the corresponding operation.
6. Display the result.
7. End the program.

---

6. Source Code

```
#include <iostream>using namespace std;
int main() {
    double num1, num2;
    char op;

    cout << "Enter first number: ";
    cin >> num1;

    cout << "Enter operator (+, -, *, /): ";
    cin >> op;

    cout << "Enter second number: ";
    cin >> num2;

    double result;

    switch(op) {
        case '+':
            result = num1 + num2;
            cout << "Result: " << result << endl;
            break;
        case '-':
            result = num1 - num2;
            cout << "Result: " << result << endl;
            break;
        case '*':
            result = num1 * num2;
            cout << "Result: " << result << endl;
            break;
        case '/':
            if (num2 != 0)
                cout << "Result: " << num1 / num2 << endl;
            else
                cout << "Error: Division by zero!" << endl;
            break;
        default:
            cout << "Invalid operator!" << endl;
    }

    return 0;
}
```

7. Compilation and Execution Steps

Step 1: Create the file

```
nano calculator.cpp
```

Step 2: Save the code and compile

```
g++ calculator.cpp -o calculator
```

Step 3: Run the program

```
./calculator
```

8. Sample Output

```
Enter first number: 12

Enter operator (+, -, *, /): *

Enter second number: 3Result: 36
```

Another Example:

```
Enter first number: 10

Enter operator (+, -, *, /): /

Enter second number: 0Error: Division by zero!
```

---

9. Result

The program successfully performs arithmetic operations and displays the correct result.
It validates user input to avoid division by zero and handles invalid operators appropriately.

---

10. Conclusion

This project demonstrates how a simple calculator can be developed using **C++** and run in a **Linux VirtualBox environment**.
It helps understand **basic syntax, operators, conditional logic, and program structure** in C++.
This project can be extended to include more advanced mathematical functions like exponentiation, modulus, and square roots.

---

11. Future Enhancements

- Add support for multiple operations in a single run (menu-driven interface).
- Include advanced mathematical functions.
- Add error handling for invalid inputs.
- Implement a graphical user interface (GUI).

---

12. References

- C++ Documentation – https://cplusplus.com/doc/tutorial
- GNU Compiler Collection – https://gcc.gnu.org
- Ubuntu Official Documentation – https://ubuntu.com/tutorials