

# **MINI PROJECT REPORT**

## **ON**

### **“ SIMPLE CALCULATOR ”**

**Submitted By :-**

**Name :-** Moirangthem Satyabrata

**UID :-** 24MCA20343

**Branch / Semester :-** MCA 1<sup>st</sup>

**Subject :-** Linux Programming

**Subject Code :-** 24CAP – 607

**Date of Performance :-** 24/10/2024

**Submitted To :-**

Mr. Navdeep Singh

**Date of Submission:** /11/2024

University Institute of Computing  
Chandigarh University, Gharuan, Mohali

## DECLARATION

We hereby declare that this submission is our own work and that, to the best of mine knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award for any other degree or diploma of the university or other institute of higher learning, except where acknowledgement has been made in the text.

Name :- Moirangthem Satyabrata

UID :- 24MCA20343

Session 2024 – 25

## **CERTIFICATE**

It is to certify that Moirangthem Satyabrata of class MCA 1st Year under University Roll Number :- 24MCA20343 has completed the project titled “ Simple Calculator ” in which the language used is HTML, CSS, PHP, MySQL for the Masters of Computer Application 1st Semester under my supervision. The work done in project is a result of the candidate's own efforts and report maintains is satisfied as per requirement.

Project Guide :-

Mr. Navdeep Singh

## INDEX

Sl.No.	Content
1	ABSTRACT
2	INTRODUCTION
3	OBJECTIVE
4	BENEFITS
5	LIMITATION
6	CODE
7	OUTPUT
8	CONCLUSION
9	GITHUB

## Abstract

The development of a simple calculator using Python in a Linux environment combines the strengths of Python's programming simplicity with Linux's powerful command-line interface. This project focuses on building a basic calculator that performs arithmetic operations such as addition, subtraction, multiplication, and division. Using Python as the core programming language, the calculator script is implemented to work efficiently within the Linux terminal, providing a quick and user-friendly solution for basic calculations. This report outlines the motivation, implementation, and practical applications of creating a Python-based calculator on Linux.

Python, known for its readability and extensive libraries, makes this project ideal for those seeking to understand fundamental programming and automation in a Linux environment. This calculator project explores essential programming concepts, including functions, conditional statements, and user input handling. By integrating Python with Linux, the project also demonstrates how command-line tools can work seamlessly with custom scripts to enhance productivity, especially in development and server environments.

The primary goal of this project is to develop a calculator application that is accessible, easy to use, and adaptable to other computational tasks. This project report discusses the calculator's functionality, limitations, and potential for expansion to support more complex operations. Ultimately, this project serves as an introductory guide to Python scripting in Linux, providing foundational skills that are essential for more advanced programming, data processing, and automation tasks.

## Introduction

Linux is widely used for its open-source architecture, security features, and efficient command-line tools, making it the preferred operating system for developers, system administrators, and data professionals. With Linux's command-line interface, users can perform a variety of tasks without the need for graphical interfaces, promoting efficiency and control. The integration of Python—a versatile, high-level programming language—with Linux offers the perfect environment to create small yet powerful applications. This report presents a project on creating a Python-based simple calculator in Linux, emphasizing the importance of basic scripting skills.

A simple calculator is a useful project to demonstrate how Python and Linux can work together to perform practical tasks. The calculator application allows users to perform essential arithmetic operations directly from the Linux terminal. By developing this calculator, users are introduced to Python programming concepts such as functions, loops, and conditionals, which form the backbone of more complex applications. The choice of Python is due to its ease of use and the readability of its syntax, making it an excellent language for beginners while still being powerful for advanced users.

Through this project, users gain a deeper understanding of Linux scripting capabilities and Python's role in task automation and scripting. The report will cover the calculator's design, its functionality, and the coding practices used to ensure efficiency and reliability in a Linux environment. This introduction to Python in Linux provides a foundation for further exploration into data processing, system automation, and application development.

## Objective

The main objective of this project is to develop a simple yet effective calculator application using Python that operates seamlessly within a Linux environment. This calculator is designed to perform basic arithmetic operations such as addition, subtraction, multiplication, and division, making it a practical tool for quick calculations. By building this calculator, the project aims to offer a straightforward, command-line solution for users, reducing the need for external calculator software in Linux.

Another objective is to introduce beginners to the essentials of Python programming within a Linux terminal. By engaging with a fundamental project, users gain confidence in handling Python functions, control statements, and error handling, all crucial for developing more complex applications. This project is structured to provide a clear understanding of how Python scripts can enhance productivity in Linux and help automate repetitive or routine calculations.

Lastly, this project intends to promote command-line proficiency and foster an appreciation for the simplicity and power of scripting in a Linux environment. By the end of this project, users will not only have a functional calculator but also a foundational skill set for creating custom scripts and applications that can streamline their workflows. This practical application of Python in Linux forms a solid base for further exploration into software development, data analysis, and systems automation.

## Technology Used

**Operating System :-** Linux (any distribution, such as Ubuntu, Fedora, or Debian)

**Programming Language :-** Python (version 3.x)

**Development Tools :-** Linux terminal, text editor (such as Vim, Nano, or VS Code), and Python's built-in libraries

**Libraries and Modules :-** Standard Python libraries, with a focus on built-in functions for arithmetic operations, handling user input, and displaying output

## Benefits :-

- 1. Lightweight and Efficient :-** The calculator is a command-line application, making it lightweight and fast. It requires minimal system resources, ideal for quick calculations in a Linux environment.
- 2. Ease of Use :-** Written in Python, the calculator code is simple and easy to understand, making it suitable for beginners and those learning Python or command-line applications on Linux.
- 3. Customizable :-** Users can modify or expand the Python code to add more advanced operations or features, allowing for customization based on specific needs.
- 4. Accessibility :-** Since the calculator is run in the Linux terminal, it can be accessed quickly without the need for additional software or a graphical user interface.



5. **Enhances Command-Line Proficiency :-** By using the calculator in a Linux environment, users gain familiarity with command-line operations and Python scripting, enhancing their proficiency in both Linux and programming.

## **Limitations :-**

1. **Limited Functionality :-** The calculator only performs basic arithmetic operations, limiting its use for more advanced mathematical calculations like trigonometry, logarithms, or exponential functions.
2. **No Graphical User Interface :-** As a command-line application, it lacks a graphical interface, which may be challenging for users who are unfamiliar with terminal operations.
3. **Error Handling :-** Basic error handling is implemented, but the calculator may still encounter issues with invalid input formats, making it less robust for varied user input.
4. **Platform Dependency :-** While Python is cross-platform, the calculator is designed specifically for Linux command-line usage, which may not be as intuitive or accessible on other operating systems without modification.
5. **Limited Input Types :-** This calculator can handle only integer and floating-point numbers. Complex numbers or symbolic math are beyond its current scope without additional libraries or modifications.

## Code

# Simple calculator in Python

```
def add(x, y):
```

```
    return x + y
```

```
def subtract(x, y):
```

```
    return x - y
```

```
def multiply(x, y):
```

```
    return x * y
```

```
def divide(x, y):
```

```
    if y == 0:
```

```
        return "Cannot divide by zero"
```

```
    return x / y
```

```
print("Select operation:")
```

```
print("1. Add")
```

```
print("2. Subtract")
```

```
print("3. Multiply")
```

```
print("4. Divide")
```

```
choice = input("Enter choice (1/2/3/4): ")

num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

if choice == '1':
    print("Result:", add(num1, num2))

elif choice == '2':
    print("Result:", subtract(num1, num2))

elif choice == '3':
    print("Result:", multiply(num1, num2))

elif choice == '4':
    print("Result:", divide(num1, num2))

else:
    print("Invalid input")
```

Activities Text Editor Nov 7 01:30

Open \*calculator.py Save

```

1 def add(x, y):
2     return x + y
3
4 def subtract(x, y):
5     return x - y
6
7 def multiply(x, y):
8     return x * y
9
10 def divide(x, y):
11     if y == 0:
12         return "Error! Division by zero."
13     return x / y
14
15 def main():
16     print("Welcome to the Simple Calculator!")
17     print("Select operation:")
18     print("1. Add")
19     print("2. Subtract")
20     print("3. Multiply")
21     print("4. Divide")
22
23     while True:
24         choice = input("Enter choice (1/2/3/4) or 'q' to quit: ")
25
26         if choice == 'q':
27             print("Exiting the calculator. Goodbye!")
28             break
29
30         if choice in ('1', '2', '3', '4'):
31             try:
32                 num1 = float(input("Enter first number: "))
33                 num2 = float(input("Enter second number: "))
34             except ValueError:
35                 print("Invalid input! Please enter numeric values.")
36                 continue
37
38             if choice == '1':
39                 print(f"{num1} + {num2} = {add(num1, num2)}")

```

Python Tab Width: 8 Ln 48, Col 2 INS

```

37
38         if choice == '1':
39             print(f"{num1} + {num2} = {add(num1, num2)}")
40         elif choice == '2':
41             print(f"{num1} - {num2} = {subtract(num1, num2)}")
42         elif choice == '3':
43             print(f"{num1} * {num2} = {multiply(num1, num2)}")
44         elif choice == '4':
45             print(f"{num1} / {num2} = {divide(num1, num2)}")
46         else:
47             print("Invalid choice! Please select a valid operation.")
48
49 if __name__ == "__main__":
50     main()

```

Python Tab Width: 8 Ln 48, Col 1 INS

## Output

```
Activities Terminal Nov 7 01:34 msatyabrata@localhost:~
[msatyabrata@localhost ~]$ python calculator.py
Welcome to the Simple Calculator!
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4) or 'q' to quit: 1
Enter first number: 5
Enter second number: 8
5.0 + 8.0 = 13.0
Enter choice (1/2/3/4) or 'q' to quit: 4
Enter first number: 9
Enter second number: 7
9.0 / 7.0 = 1.2857142857142858
Enter choice (1/2/3/4) or 'q' to quit: q
Exiting the calculator. Goodbye!
[msatyabrata@localhost ~]$
```

## Conclusion

This project demonstrates how a simple calculator can be effectively built using Python in a Linux environment. By implementing basic arithmetic operations, the calculator offers a lightweight, accessible solution for quick calculations directly within the Linux terminal. This project not only underscores Python's versatility but also highlights the potential of command-line applications in a Linux setting, where lightweight tools are highly valued.

The development of this calculator application has provided insights into essential programming concepts, including functions, conditionals, and error handling, making it a valuable learning tool for those new to Python or Linux. While the calculator is limited in scope, its simplicity and ease of customization offer a foundation for more complex projects, encouraging further exploration into Python programming and Linux command-line scripting. Overall, this project illustrates the power of combining Linux and Python for efficient, functional applications and serves as a stepping stone for those interested in building custom command-line tools.

## GitHub

**GitHub Link :-**