Training Report – Day 8

Topic Covered Today:

- Writing Python programs using Vim editor
- Creating a **simple calculator program** in Python
- Running and testing Python scripts from the terminal

Key Learning:

Using Vim for Python Programming:

Today I learned how to use **Vim editor** not only for editing text but also for writing and saving Python programs. I practiced the workflow:

1. Open a new Python file:
2. `vim calculator.py`
3. Enter **insert mode** (`i`) and write Python code.
4. Save the file with `:w` and quit with `:q`.
5. Run the program in terminal using:
6. `python3 calculator.py`

This helped me connect Linux skills (Vim, terminal) with Python programming.

Calculator in Python:

I created a **basic calculator program** that can perform operations like addition, subtraction, multiplication, and division.

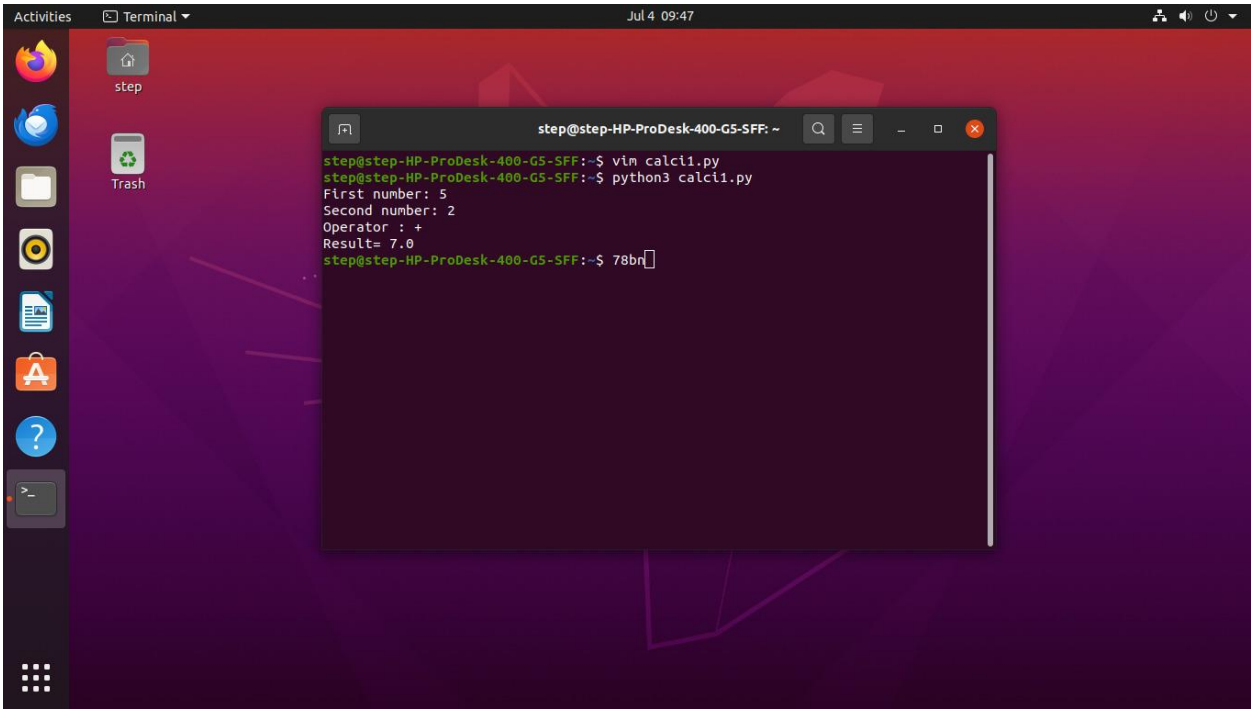
Sample code written in Vim:

```
# Simple Calculator Program

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 x / y
    else:
        return "Error! Division by zero."

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")
```

This program takes two numbers and an operator choice from the user, then prints the result.

Output:



Activities / Assignments:

- Created a Python file (`calculator.py`) using Vim editor.
- Wrote a simple calculator program inside Vim.
- Saved and executed the program directly from the terminal.
- Tested different cases like addition, subtraction, multiplication, division, and division by zero.
- Modified the code to handle errors (e.g., invalid choices).

Personal Reflection for Day 8:

Today's session was very practical because it combined **Linux, Vim, and Python programming** together. Initially, writing code in Vim was slightly harder than using modern editors like VS Code or Jupyter, but I realized Vim is very powerful once I got comfortable with the commands.

Creating a calculator program gave me confidence in Python basics like functions, user input, conditionals (`if-else`), and error handling. Running the code from the terminal showed me how developers work in real Linux environments without depending on graphical tools.

Overall, today's practice strengthened my coding skills and made me more comfortable with integrating Linux tools and Python program.