EXPERIMENT NO 1

Title : Write a python program for implementation of calculator.

Objectives : 1. To understand the basics of Python programming.

2. To implement fundamental arithmetic operations.

3. To create a user-friendly calculator interface.

Key Concepts: 1. Variables and data types

2. Basic arithmetic operations

3. Input/output handling.

4. Conditional statements.

5. Looping constructs.

Theory –

What is Python –

Python is a high-level, interpreted programming language known for its simplicity and readability. It is versatile and widely used in various fields such as web development, data science, artificial intelligence, scientific computing, automation, and more.

Why to use Python –

- **Readability**: Python's syntax is designed to be readable and concise, which reduces the cost of program maintenance and development.
- **Versatility**: Python supports multiple programming paradigms and has a vast ecosystem of libraries and frameworks for various applications.
- **Portability**: Python is platform-independent, meaning code written in Python can run on various platforms without modification.
- **Interpreted Language**: Python is an interpreted language, allowing for rapid development and prototyping.
- **Integration**: Python easily integrates with other languages and tools, making it suitable for building complex systems and integrating with existing infrastructure.

Variables and Data Types in Python:

Variables are containers for storing data values. In Python, you can create variables by assigning values to them using the = operator.

Data types define the type of data that a variable can store. Common data types in Python include integers (int), floating-point numbers (float), strings (str), booleans (bool), etc.

➢ **Basic Arithmetic Operations:**

Addition (+), subtraction (-), multiplication (*), and division (/) are fundamental arithmetic operations used for performing mathematical calculations.

Modulus (%) operator returns the remainder of the division between two numbers.

Square root can be calculated using the math.sqrt() function from the math module.

➢ **Input/Output Handling:**

Input handling involves accepting user input, which can be done using functions like input() in Python. In the program, input() is used to get user input for numbers and operation selection.

Output handling involves displaying information to the user. In the program, print() function is used to display messages and results to the user.

➢ **Conditional Statements:**

Conditional statements are used to execute different code blocks based on specified conditions. In the program, if, elif, and else statements are used to determine the selected operation and handle error conditions.

➢ **Looping Constructs:**

Looping constructs allow repetitive execution of a block of code. In the program, a while loop is used to continuously prompt the user for input until they choose to exit the calculator.

➢ Python Function:

- This Python function is named add and it takes two parameters, x and y. Inside the function, it simply adds these two parameters together using the + operator and returns the result.
- Here's a breakdown:
- def add(x, y):: This line defines the function named add which takes two parameters, x and y.
- return x + y: This line calculates the sum of x and y using the + operator, and then returns the result.
- When you call this function, you would pass two arguments, for example, add(3, 5) would return 8 because 3 + 5 = 8

Algorithm step by step: -

1. **Display a Menu:**

    Show the user a menu with options for different operations (addition, subtraction, multiplication, division, modulus, square root).

2. **Prompt User for Operation:**

    Ask the user to choose an operation from the menu. This can be done by displaying a message and accepting input from the user.

3. **Check Validity of Operation:**

    Verify if the user's input corresponds to a valid operation option. This can be done by checking if the input matches one of the predefined operation choices.

4. **Prompt User for Numbers:**

    If the operation is valid, prompt the user to enter the required numbers for the chosen operation. For example, if the user selects addition, prompt them to enter two numbers to add.

5. **Perform Selected Operation:**

    Based on the user's choice and input numbers, perform the selected operation (addition, subtraction, etc.). This involves executing the corresponding code logic for each operation.

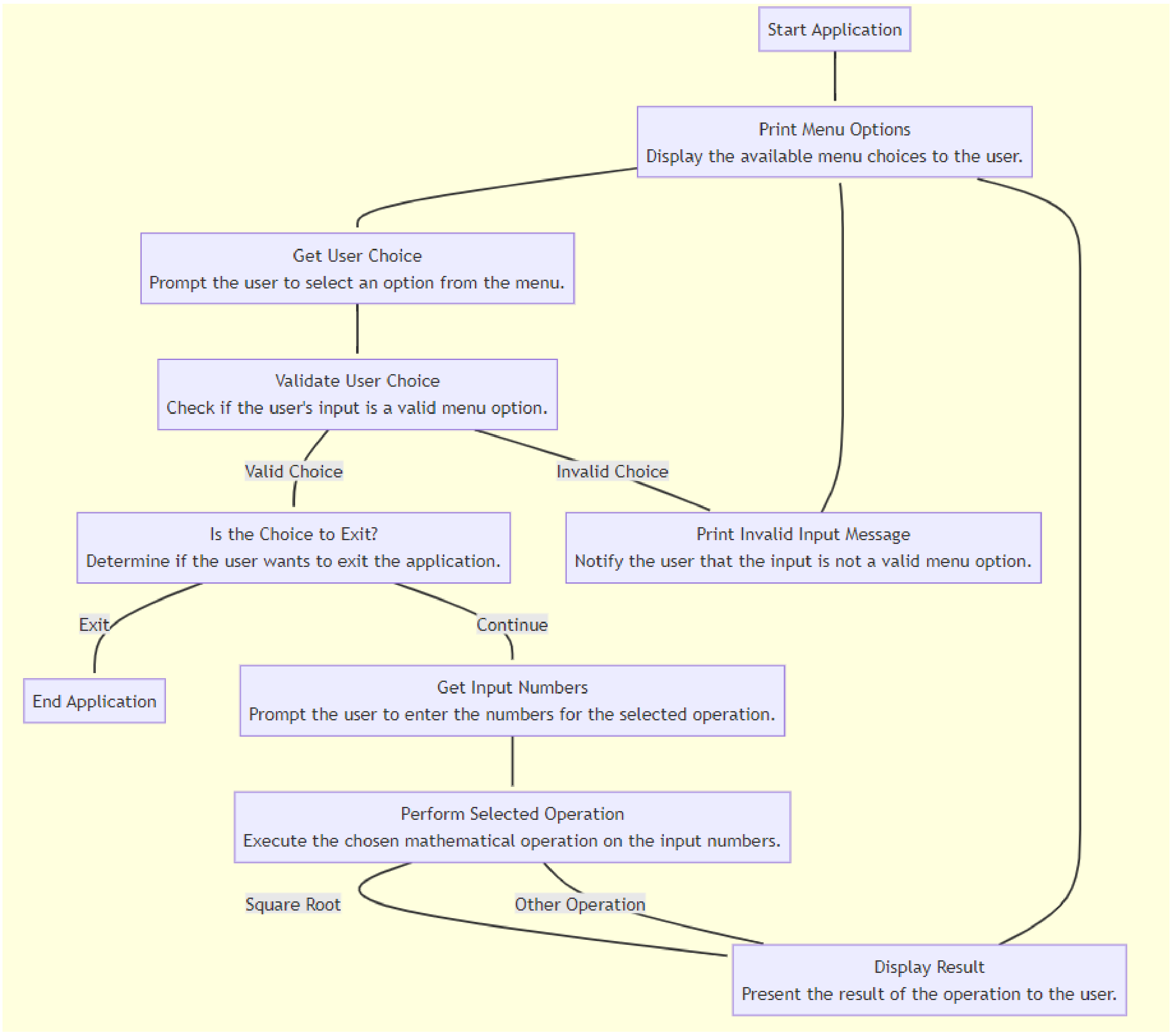
6. **Display Result:**

    After performing the operation, display the result to the user. This can be done by printing the result to the console or displaying it in a message box.

7. **Allow User to Continue or Exit:**

    Provide the user with the option to continue using the calculator or to exit the program. This can be achieved by presenting the user with a choice to either repeat the process from step 2 or to exit the program.

Flowchart:



Start Application
Print Menu Options
Display the available menu choices to the user.
Get User Choice
Prompt the user to select an option from the menu.
Validate User Choice
Check if the user's input is a valid menu option.
Valid Choice
Invalid Choice
Is the Choice to Exit?
Determine if the user wants to exit the application.
Print Invalid Input Message
Notify the user that the input is not a valid menu option.
Exit
Continue
End Application
Get Input Numbers
Prompt the user to enter the numbers for the selected operation.
Perform Selected Operation
Execute the chosen mathematical operation on the input numbers.
Square Root
Other Operation
Display Result
Present the result of the operation to the user.

Program:

```
import math

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

def modulus(x, y):
    return x % y

def square_root(x):
    return math.sqrt(x)

def calculator():
    while True:
        print("Select operation:")
        print("1. Add")
        print("2. Subtract")
        print("3. Multiply")
        print("4. Divide")
        print("5. Modulus")
        print("6. Square Root")
        print("7. Exit")

        choice = input("Enter choice (1/2/3/4/5/6/7): ")

        if choice in ('1', '2', '3', '4', '5', '6'):
            num1 = float(input("Enter first number: "))
            if choice != '6':
                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))
            elif choice == '5':
                print("Result:", modulus(num1, num2))
            elif choice == '6':
                print("Result:", square_root(num1))
        elif choice == '7':
            print("Exiting calculator. Goodbye!")
            break
        else:
                print("Invalid input")


        calculator()
```

Conclusion :

In this experiment, we successfully implemented a simple calculator program in Python. We learned about basic arithmetic operations, handling user input, and creating a menu-driven interface. This exercise provided a solid foundation for understanding Python programming concepts and practical application.