NAME - AVIRAL JAIN

PROJECT - CALCULATOR (BASIC TO ADVANCE FEATURES)

Step 1: Basic Calculator

Create a Python program that prompts the user for two numbers and an operator (+, -, *, /). Perform the corresponding arithmetic operation based on the operator and display the result to the user.

In [*]:
```
def calculator():
    num1 = float(input("Enter the first number: "))
    operator = input("Enter an operator (+, -, *, /): ")
    num2 = float(input("Enter the second number: "))

    if operator == '+':
        result = num1 + num2
    elif operator == '-':
        result = num1 - num2
    elif operator == '*':
        result = num1 * num2
    elif operator == '/':
        result = num1 / num2
    else:
        print("Invalid operator!")
        return

    print("Result:", result)

calculator()
```

Step 2: Input Validation

Add input validation to handle invalid input from the user, such as non-numeric values or unsupported operators. Display an error message and prompt the user to enter valid input.

In [*]:
```
def calculator():
    try:
        num1 = float(input("Enter the first number: "))
        operator = input("Enter an operator (+, -, *, /): ")
        num2 = float(input("Enter the second number: "))

        if operator == '+':
            result = num1 + num2
        elif operator == '-':
            result = num1 - num2
        elif operator == '*':
            result = num1 * num2
        elif operator == '/':
            result = num1 / num2
        else:
            print("Invalid operator!")
            return

        print("Result:", result)
    except ValueError:
        print("Invalid input!")
calculator()
```

Step 3: Continuous Calculations

Add a loop to allow the user to perform multiple calculations without exiting the program. Prompt the user if they want to continue or exit after each calculation.

In [*]:
```
def calculator():
    while True:
        try:
            num1 = float(input("Enter the first number: "))
            operator = input("Enter an operator (+, -, *, /) or 'q' to quit: ")

            if operator == 'q':
                print("Exiting calculator...")
                break

            num2 = float(input("Enter the second number: "))

            if operator == '+':
                result = num1 + num2
            elif operator == '-':
                result = num1 - num2
            elif operator == '*':
                result = num1 * num2
            elif operator == '/':
                result = num1 / num2
            else:
                print("Invalid operator!")
                continue

            print("Result:", result)
        except ValueError:
            print("Invalid input!")

calculator()
```

Step 4: Error Handling

Implement error handling to handle exceptional situations, such as division by zero. Display appropriate error messages and handle such cases gracefully.

In [*]:
```
def calculator():
    while True:
        try:
            operator = input("Enter an operator (+, -, *, /) or 'q' to quit: ")
            if operator == 'q':
                print("Exiting calculator...")
                quit()

            num1 = float(input("Enter the first number: "))

            num2 = float(input("Enter the second number: "))
            result = 0
            if operator == '+':
                result = num1 + num2
            elif operator == '-':
                result = num1 - num2
            elif operator == '*':
                result = num1 * num2
            elif operator == '/':
                if num2 == 0:
                    print("Error: Division by zero!")
                    exit()
                result = num1 / num2
            else:
                print("Invalid operator!")

            print("Result:", result)
        except ValueError:
            print("Invalid input!")

calculator()
```

Step 5: Advanced Operations

Expand the calculator to support more advanced operations like exponentiation (^), square root, and modulus (%).

In [*]:
```
import math

def calculator():
    while True:
        try:
            num1 = float(input("Enter the first number: "))
            operator = input("Enter an operator (+, -, *, /, ^, sqrt, %) or 'q' to quit: ")

            if operator == 'q':
                print("Exiting calculator...")
                break

            if operator == 'sqrt':
                result = math.sqrt(num1)
            else:
                num2 = float(input("Enter the second number: "))

                if operator == '+':
                    result = num1 + num2
                elif operator == '-':
                    result = num1 - num2
                elif operator == '*':
                    result = num1 * num2
                elif operator == '/':
                    if num2 == 0:
                        print("Error: Division by zero!")
                        continue
                    result = num1 / num2
                elif operator == '^':
                    result = num1 ** num2
                elif operator == '%':
                    result = num1 % num2
                else:
                    print("Invalid operator!")
                    continue

            print("Result:", result)
        except ValueError:
            print("Invalid input!")
calculator()
```

Step 6: User-Friendly Interface

Improve the user interface by displaying a menu of available operations and allowing the user to select their desired operation. Provide clear instructions and error messages to enhance the user experience.

In [*]:
```
import math

def calculator():
    while True:
        try:
            print("Calculator Menu:")
            print("1. Addition (+)")
            print("2. Subtraction (-)")
            print("3. Multiplication (*)")
            print("4. Division (/)")
            print("5. Exponentiation (^)")
            print("6. Square Root (sqrt)")
            print("7. Modulus (%)")
            print("q. Quit")

            choice = input("Select an operation (1-7) or 'q' to quit: ")

            if choice == 'q':
                print("Exiting calculator...")
                break

            if choice not in ['1', '2', '3', '4', '5', '6', '7']:
                print("Invalid choice!")
                continue

            num1 = float(input("Enter the first number: "))

            if choice != '6':
                num2 = float(input("Enter the second number: "))

            if choice == '1':
                result = num1 + num2
            elif choice == '2':
                result = num1 - num2
            elif choice == '3':
                result = num1 * num2
            elif choice == '4':
                if num2 == 0:
                    print("Error: Division by zero!")
                    continue
                result = num1 / num2
            elif choice == '5':
                result = num1 ** num2
            elif choice == '6':
                result = math.sqrt(num1)
            elif choice == '7':
                result = num1 % num2

            print("Result:", result)
        except ValueError:
            print("Invalid input!")
```

```
calculator()
```

Calculator Program

This Jupyter Notebook contains the code for a calculator program implemented in Python. The calculator provides both basic and scientific calculator functionalities. It allows users to perform arithmetic operations, store values in memory, recall values from memory, and view calculation history.

To run the calculator program, execute the code cells below. Follow the prompts to perform calculations, navigate the menu options, and interact with the calculator.

Let's get started!

Step 1: Import the math module

Import the math module to access mathematical functions required for scientific calculations.

In [*]:
```
import math
```

Step 2: Create a Calculator Class A class named Calculator is defined to encapsulate the calculator functionalities. The **init** method initializes the history list to store calculation history and the memory variable to store memory values.

In [*]:
```
class Calculator:
    def __init__(self):
        self.history = []
        self.memory = 0
```

Step 3: Implement add_to_history Method The add_to_history method is created to add the expression and result of a calculation to the history list.

In [*]:
```
    def add_to_history(self, expression, result):
        self.history.append((expression, result))
```

Step 4: Implement display_history Method The display_history method is implemented to display the calculation history stored in the history list.

In [*]:
```
    def display_history(self):
        print("Calculation History:")
        for index, (expression, result) in enumerate(self.history, 1):
            print(f"{index}. {expression} = {result}")
        print()
```

Step 5: Implement clear_history Method The clear_history method is created to clear the calculation history by resetting the history list

In [*]:
```
    def clear_history(self):
        self.history = []
```

Step 6: Implement store_to_memory Method The store_to_memory method is implemented to store a given value in the memory variable.

In [*]:
```
    def store_to_memory(self, value):
        self.memory = value
        print(f"Stored value {value} to memory.")
```

Step 7: Implement recall_memory Method The recall_memory method is created to display the value stored in the memory variable.

In [*]:
```
    def recall_memory(self):
        print(f"Memory value: {self.memory}")
```

Step 8: Implement clear_memory MethodThe clear_memory method is implemented to reset the memory variable to 0.

In [*]:
```
    def clear_memory(self):
        self.memory = 0
        print("Memory cleared.")
```

Step 9: Implement calculator Method The calculator method is created to handle the calculator operations. It displays a menu with options for the basic calculator, scientific calculator, viewing history, and memory operations. Based on the user's choice, it performs the corresponding operation.

In [*]:
```
    def calculator(self):
        while True:
            # Display menu options and handle user input
```

Step 10: Implement Basic Calculator Operations Within the calculator method, the basic calculator operations are implemented. It prompts the user to enter the first number, operator, and second number for basic calculations. It performs the arithmetic operation based on the operator and displays the result. The expression and result are stored in the history using the add_to_history method.

In [*]:
```
            num1 = float(input("Enter the first number: "))
            operator = input("Enter an operator (+, -, *, /) or 'q' to quit: ")
            # Handle basic calculator operations
```

Step 11: Implement Scientific Calculator Operations Within the calculator method, the scientific calculator operations are implemented. It displays a menu with options for square root, sine, cosine, and tangent. The user selects an option and enters a number for the scientific operation. The selected scientific operation is performed using the appropriate math function from the math module. The result of the scientific calculation is displayed, and the expression and result are stored in the history using the add_to_history method.

In [*]:
```
            print("Scientific Calculator Mode:")
            scientific_choice = input("Select an option (1-4) or any other key to go back: ")
            # Handle scientific calculator operations
```

Step 12: Run the Calculator An instance of the Calculator class is created, and the calculator method is called to start the calculator program.

In [*]:
```
calculator = Calculator()
calculator.calculator()
```

From 1 to 12 step in one cell

In [*]:
```
import math

class Calculator:
    def __init__(self):
        self.history = []
        self.memory = 0

    def add_to_history(self, expression, result):
        self.history.append((expression, result))

    def display_history(self):
        print("Calculation History:")
        for index, (expression, result) in enumerate(self.history, 1):
            print(f"{index}. {expression} = {result}")
        print()

    def clear_history(self):
        self.history = []

    def store_to_memory(self, value):
        self.memory = value
        print(f"Stored value {value} to memory.")

    def recall_memory(self):
        print(f"Memory value: {self.memory}")

    def clear_memory(self):
        self.memory = 0
        print("Memory cleared.")

    def calculator(self):
        while True:
            print("Calculator Menu:")
            print("1. Basic Calculator")
            print("2. Scientific Calculator")
            print("3. History")
            print("4. Memory")
            print("q. Quit")

            choice = input("Select an option (1-4) or 'q' to quit: ")

            if choice == 'q':
                print("Exiting calculator...")
                break

            if choice not in ['1', '2', '3', '4']:
                print("Invalid choice!")
                continue

            if choice == '3':
                self.display_history()
                continue
```

```
            if choice == '4':
                print("Memory Menu:")
                print("1. Store to Memory")
                print("2. Recall Memory")
                print("3. Clear Memory")

                memory_choice = input("Select an option (1-3) or any other key to go back: ")

                if memory_choice == '1':
                    value = float(input("Enter a value to store in memory: "))
                    self.store_to_memory(value)
                elif memory_choice == '2':
                    self.recall_memory()
                elif memory_choice == '3':
                    self.clear_memory()
                else:
                    continue

                print()
                continue

            num1 = float(input("Enter the first number: "))

            if choice == '1':
                operator = input("Enter an operator (+, -, *, /): ")
                num2 = float(input("Enter the second number: "))

                if operator == '+':
                    result = num1 + num2
                elif operator == '-':
                    result = num1 - num2
                elif operator == '*':
                    result = num1 * num2
                elif operator == '/':
                    if num2 == 0:
                        print("Error: Division by zero!")
                        continue
                    result = num1 / num2
                else:
                    print("Invalid operator!")
                    continue

            elif choice == '2':
                print("Scientific Calculator Mode:")
                print("1. Square Root")
                print("2. Sine")
                print("3. Cosine")
                print("4. Tangent")

                scientific_choice = input("Select an option (1-4) or any other key to go back: ")

                if scientific_choice == '1':
                    result = math.sqrt(num1)
                elif scientific_choice == '2':
```

```
                    result = math.sin(math.radians(num1))
                elif scientific_choice == '3':
                    result = math.cos(math.radians(num1))
                elif scientific_choice == '4':
                    result = math.tan(math.radians(num1))
                else:
                    continue

            expression = f"{num1} {operator} {num2}" if choice == '1' else f"{scientific_choice}({num1})"

            print("Result:", result)
            self.add_to_history(expression, result)
            print()

calculator = Calculator()
calculator.calculator()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: