

***Name: Muhammad Zain***

***Roll No: SU92-BSAIM-F24-092***

**Dynamic Calculator**

**Code Explanation**

This program is a **dynamic calculator** written in Python. The main purpose of the code is to take mathematical expressions directly from the user, solve them, and show the result. It is called “dynamic” because the user can enter any type of valid expression at runtime instead of having fixed operations in the program.

**Step 1: Importing the Required Library**

Import re

Here, we import the **re (regular expressions)** library. It helps in identifying and replacing certain patterns in the user’s input. For example, it is used to insert multiplication signs in places where the user does not write them explicitly.

**Step 2: Starting the Calculator**

print(" My Calculator ")

print("Type 'exit' to quit\n")

These lines simply display a welcome message for the calculator and guide the user that typing **exit** will close the program.

**Step 3: Loop for Continuous Input**

while True:

expression = input("Enter expression: ")

if expression.lower() == "exit":

break

* The program runs in a while True loop, which means it will keep running until the user decides to quit.
* The user types a mathematical expression as input.
* If the user types “exit” (in any case: EXIT, Exit, exit), the loop breaks and the program stops.

**Step 4: Converting Symbols**

expression = expression.replace("×", "\*").replace("÷", "/")

Since Python understands \* for multiplication and / for division, but many users write × and ÷, this line automatically converts them into Python-friendly symbols.

**Step 5: Handling Implicit Multiplication**

expression = re.sub(r'(\d)(\()', r'\1\*(', expression)

expression = expression.replace(")(", ")\*(")

In normal mathematics, people often write multiplication without a \*. For example:

* 2(3+4) means 2 \* (3+4)
* (2+3)(4-1) means (2+3) \* (4-1)

Python does not accept this format by default.  
So these two lines fix the issue.

**Step 6: Evaluating the Expression**

try:

result = eval(expression)

print("Result:", result)

except Exception as e:

print("Invalid Expression!")

* The program uses Python’s built-in eval() function to calculate the final answer.
* If the expression is valid, the result is shown on the screen.
* If the user makes a mistake (for example: unbalanced brackets or wrong symbols), the program catches the error and prints **“Invalid Expression!”** instead of crashing.

