Python CLI Calculator

This is a simple command-line calculator application built in Python. It allows users to perform basic arithmetic operations like addition, subtraction, multiplication, and division using a simple menu-based interface.

Features

* Supports Addition, Subtraction, Multiplication, Division.
* Input validation to prevent crashes on invalid inputs.
* Division by zero protection.
* Menu-driven CLI (Command Line Interface).
* Uses functions to keep the code modular and organized.
* Runs continuously in a loop until the user exits.

How to Run

1. Make sure you have Python installed.

2. Save the file as Calculator.py

3. Open your terminal or PowerShell.

4. Navigate to the folder where the file is located:

bash

cd path\to\your\folder

5. Run the code

python Calculator.py

(If you're using PowerShell, you might need to use python .\Calculator.py)

1. Function Definitions:

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."

return x / y

* These are the four basic math operations.
* Each is defined as a separate function to keep the code modular and reusable.
* The divide() function includes an error check to prevent division by zero.

2. Main Calculator Loop

def calculator():

print("Simple CLI Calculator")

print("----------------------")

while True:

# Menu Options

print("\nSelect operation:")

print("1. Add")

print("2. Subtract")

print("3. Multiply")

print("4. Divide")

print("5. Exit")

* The calculator() function runs the whole CLI experience.
* It starts with a printed header.
* Then it enters an infinite while True: loop that only ends when the user chooses to exit.

3. User Input and Validation

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

if choice == '5':

print("Exiting the calculator. Goodbye!")

break

if choice not in ('1', '2', '3', '4'):

print("Invalid choice. Please select a valid option.")

continue

* The user selects an operation.
* If they choose '5', the loop breaks and the app exits.
* Invalid menu inputs are handled with a warning and the loop continues.

4. Number Input with Error Handling

try:

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

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

except ValueError:

print("Invalid input. Please enter numeric values.")

continue

* The user inputs two numbers.
* float() is used so that decimal numbers are supported.
* A try/except block catches non-numeric inputs to avoid crashing.

5. Performing the Operation

if choice == '1':

print(f"Result: {add(num1, num2)}")

elif choice == '2':

print(f"Result: {subtract(num1, num2)}")

elif choice == '3':

print(f"Result: {multiply(num1, num2)}")

elif choice == '4':

print(f"Result: {divide(num1, num2)}")

* Based on the user's choice, the corresponding function is called and the result is printed.

6. Main Entry Point

if \_name\_ == "\_main\_":

calculator()

* This ensures that calculator() runs only when the script is executed directly (not when imported).

Sample output when run in terminal using powershell:

A black background with white text

AI-generated content may be incorrect.

Run the python file based on the file name. Make sure it’s in the correct directory.

A screenshot of a computer

AI-generated content may be incorrect.

The option displays for user to select operation. Choose either Add, Subtract, Multiply, Divide or Exit.

So I chose option 1 to add. My first number is 12 and my second number is 7 so my result is 19.0.

A screenshot of a computer program

AI-generated content may be incorrect.

Next, I chose option 2 to subtract. My first number is 9 and my second number is 8 so my result is 1.0.

A screenshot of a computer program

AI-generated content may be incorrect.

Next, I chose option 3 to multiply. My first number is 4 and my second number is 5 so my result is 20.0.

A screenshot of a computer program

AI-generated content may be incorrect.

Next, I chose option 4 to divide. My first number is 8 and my second number is 4 so my result is 2.0.

A screenshot of a computer program

AI-generated content may be incorrect.

Next, I chose option 5 to exit. So, it prints Exiting the calculator. Goodbye!

INTERVIEW QUESTIONS - PYTHON DEVELOPER INTERNSHIP

1. What is normalization?

Organizing data to reduce redundancy and improve integrity.

2. Explain primary vs foreign key.

Primary key uniquely identifies records; foreign key links to a primary key in another table.

3. What are constraints?

Rules that enforce data integrity (e.g. NOT NULL, UNIQUE).

4. What is a surrogate key?

An artificially generated key used to uniquely identify a record.

5. How do you avoid data redundancy?

Normalize data, enforce foreign keys, and avoid data duplication.

6. What is ER diagram?

A visual model of entities and their relationships in a database.

7. What are the types of relationships in DBMS?

One-to-One, One-to-Many, Many-to-One, Many-to-Many.

8. Explain the purpose of AUTO\_INCREMENT.

Automatically generates a unique ID for new rows in a table.

9. What is the default storage engine in MySQL?

InnoDB.

10. What is a composite key?

A primary key made from multiple columns together.