

Python_Activity16

Activity 1: Fix the Errors (Debugging Basics)

Problem Statement:

A new intern wrote the following Python script, but it has **syntax and logical errors**. Your job is to **fix the errors**, so the program runs correctly.

```
print("Welcome to the Python calculator!")

def add_numbers(a, b)
    return a + b

num1 = input("Enter first number: ")
num2 = input("Enter second number: ")

print("Sum: " + add_numbers(num1, num2))
```

Activity 2: Improve Readability & Best Practices

Problem Statement:

The following code works correctly but is **hard to read and not written in a structured way**. Your task is to **rewrite it using coding best practices**.

```
def f(x,y):return x*y
a=input("Enter a number:")
b=input("Enter another number:")
print(f(a,b))
```

Activity 3: Simple Error Handling for Input Validation

Problem Statement:

A developer created a script to divide two numbers, but it **crashes if a user enters non-numeric input or tries to divide it by zero**. Your task is to **add error handling** to prevent these issues.

```
def divide(a, b):  
    return a / b  
  
num1 = input("Enter numerator: ")  
num2 = input("Enter denominator: ")  
  
print("Result:", divide(num1, num2))
```

Activity 4: Refactoring for Modularity & Reusability (Intermediate Level)

Problem Statement:

A junior developer has written a simple calculator program, but it's **repetitive and difficult to modify**. Your task is to **refactor the code** to make it more **modular, reusable, and easier to maintain**.

```
num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

choice = input("Choose operation (+, -, *, /): ")

if choice == "+":
    print("Result:", num1 + num2)
elif choice == "-":
    print("Result:", num1 - num2)
elif choice == "*":
    print("Result:", num1 * num2)
elif choice == "/":
    print("Result:", num1 / num2)
else:
    print("Invalid operation!")
```

Note:

- *Give me your best output*
- *Add what you think is needed*