TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Musfira Ahmed

Intern ID: TN/IN01/PY/002

Email ID : ahmedmusfira3@gmail.com

Internship Domain : Python Development

Task Week : 03

Instructor Name : Mr Hassan Ali

Task 1 :

Create a calculator that accepts two numbers and an operator (+, -, *, /, %, //, **).

Perform the operation and display the result.

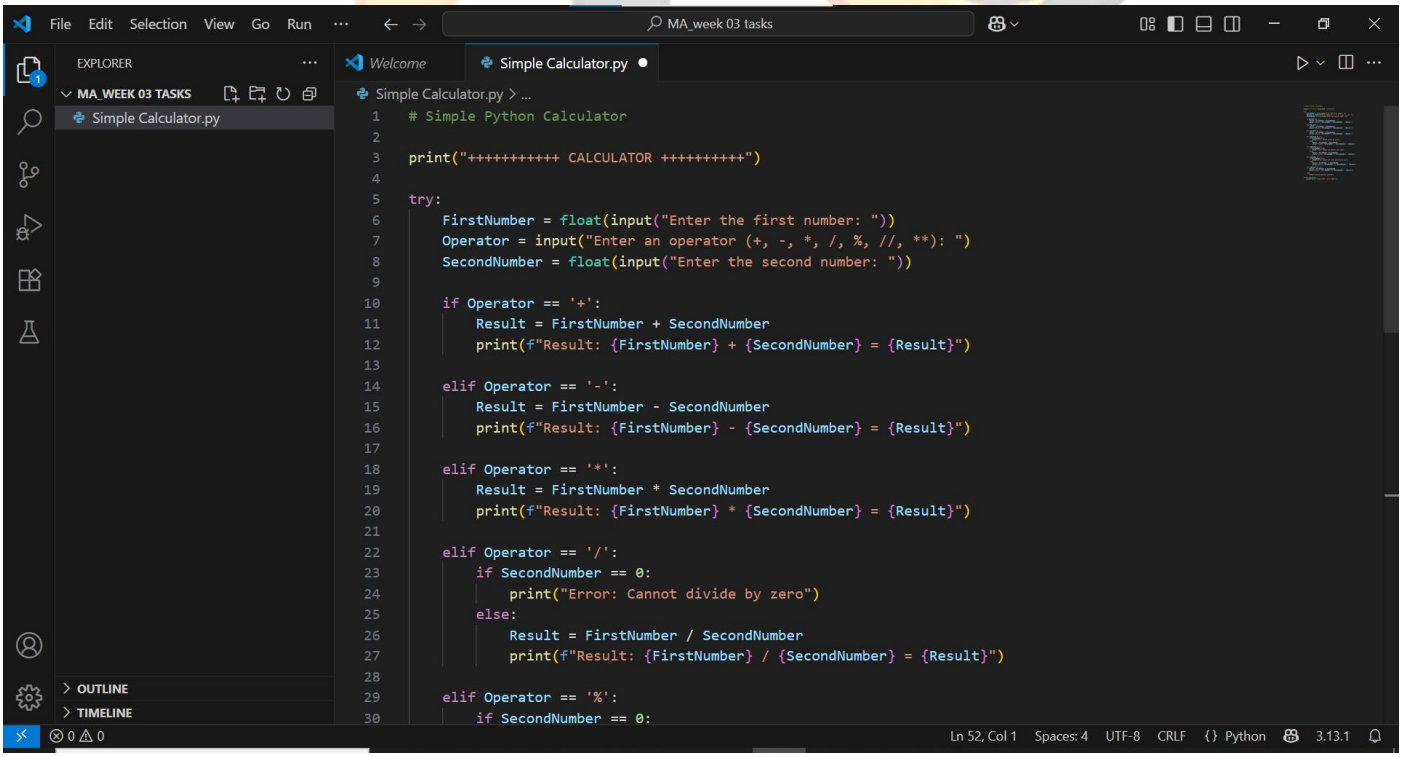
Handle division by zero safely.

Solution :

What I Did (Step by Step):

- Took two numbers and an operator as input from the user.
- Used if-elif and else statements to perform the selected operation.
- Checked for division, modulus, and floor division by zero.
- Displayed the result or appropriate error message.

Code Screenshots




```python
# Simple Python Calculator

print("++++++++++ CALCULATOR ++++++++++")

try:
    FirstNumber = float(input("Enter the first number: "))
    Operator = input("Enter an operator (+, -, *, /, %, //, **): ")
    SecondNumber = float(input("Enter the second number: "))

    if Operator == '+':
        Result = FirstNumber + SecondNumber
        print(f"Result: {FirstNumber} + {SecondNumber} = {Result}")

    elif Operator == '-':
        Result = FirstNumber - SecondNumber
        print(f"Result: {FirstNumber} - {SecondNumber} = {Result}")

    elif Operator == '*':
        Result = FirstNumber * SecondNumber
        print(f"Result: {FirstNumber} * {SecondNumber} = {Result}")

    elif Operator == '/':
        if SecondNumber == 0:
            print("Error: Cannot divide by zero")
        else:
            Result = FirstNumber / SecondNumber
            print(f"Result: {FirstNumber} / {SecondNumber} = {Result}")

    elif Operator == '%':
        if SecondNumber == 0:
```

```python
    elif Operator == '%':
        if SecondNumber == 0:
            print("Error: Cannot use modulus with zero")
        else:
            Result = FirstNumber % SecondNumber
            print(f"Result: {FirstNumber} % {SecondNumber} = {Result}")

    elif Operator == '//':
        if SecondNumber == 0:
            print("Error: Cannot use floor division by zero")
        else:
            Result = FirstNumber // SecondNumber
            print(f"Result: {FirstNumber} // {SecondNumber} = {Result}")

    elif Operator == '**':
        Result = FirstNumber ** SecondNumber
        print(f"Result: {FirstNumber} ** {SecondNumber} = {Result}")

    else:
        print("Invalid operator entered")

except ValueError:
    print("Error: Please enter valid numbers")
```

Output Screenshot

```
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/Simple Calculator.py"
++++++++++ CALCULATOR ++++++++++
Enter the first number: 4
Enter an operator (+, -, *, /, %, //, **): *
Enter the second number: 5
Result: 4.0 * 5.0 = 20.0
PS D:\MA_week 03 tasks>
```

Learning and Challenges:

- Learned basic arithmetic operations and conditional logic.
- Faced issues with division by zero and handled them using conditions.
- Understood how to convert inputs and print results clearly.
- Practiced simple error handling with try-except for invalid input.

Task 02:

Take marks of 3 subjects.

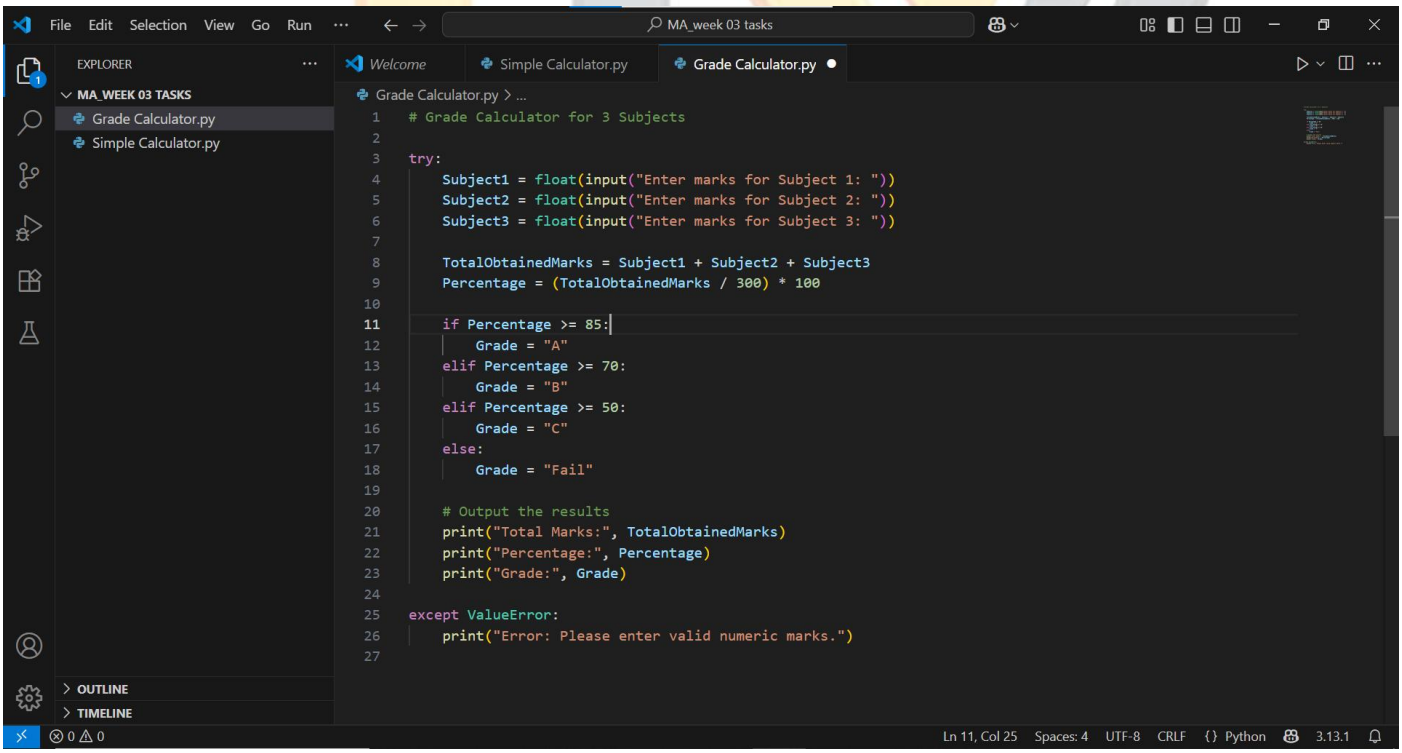
Calculate total, percentage and assign grade:

A (>=85), B (>=70), C (>=50), Fail (<50).

What I Did (Step by Step):

- Took marks for 3 subjects as input from the user.
- Calculated the total and percentage assuming each subject is out of 100.
- Used if-elif statements to assign grades based on percentage.
- Displayed total, percentage, and grade with proper error handling.

Code Screenshots



Output Screenshot

```
PROBLEMS   OUTPUT   TERMINAL   PORTS
TERMINAL                                                    Python
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/Grade Calculator.py"
Enter marks for Subject 1: 78
Enter marks for Subject 2: 76
Enter marks for Subject 3: 89
Total Marks: 243.0
Percentage: 81.0
Grade: B
PS D:\MA_week 03 tasks>
Ln 18, Col 23   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges:

. Learned how to work with multiple inputs and calculations.

. Practiced using conditions to categorize values (grade logic).

. Faced minor issues with invalid input and fixed them using try-except.

. Understood how to structure output for better readability.

Task 03:

Ask user for monthly income and expenses.

Calculate savings and classify:

>10000 = Saving Well, 5000–9999 = Average, <5000 = Try to Save.

What I Did (Step by Step):

· Took income and expense values as input from the user.

· Subtracted expenses from income to calculate savings.

· Classified savings using if-elif conditions.

· Displayed the savings amount and status.

Code Screenshots

```python
# Monthly Savings Classification

try:
    Income = float(input("Enter your monthly income: "))
    Expenses = float(input("Enter your monthly expenses: "))

    Savings = Income - Expenses

    if Savings > 10000:
        Status = "Saving Well"

    elif 5000 <= Savings <= 9999:

        Status = "Average"

    elif Savings < 5000:

        Status = "Try to Save"

    else:
        Status = "Not Classified!"

    print("Your Savings are :", Savings)
    print("Status:", Status)

except ValueError:
    print("Error: Please enter valid numeric values.")
```

Output Screenshot

```
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/Monthly income and expense calculator.py"
Enter your monthly income: 500000
Enter your monthly expenses: 10000
Your Savings are : 490000.0
Status: Saving Well
PS D:\MA_week 03 tasks>
```

Learnings and Challenges:

- Learned how to use basic arithmetic and conditional logic in real-life scenarios.
- Faced input validation issues and handled them with try-except.
- Practiced displaying results in a clear and readable format.
- Understood how thresholds can be used to categorize user input

Task 04:

Build a login system. Ask username & password.

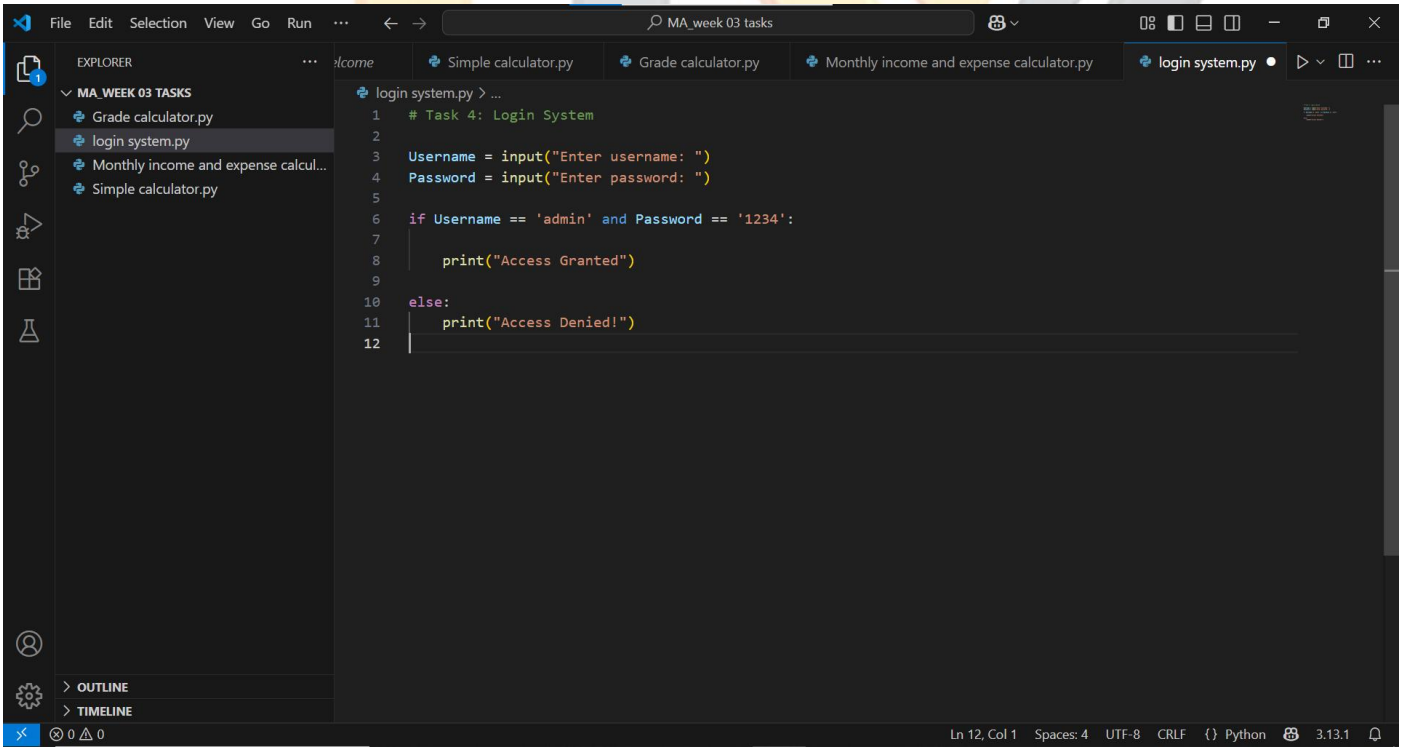
If **username = 'admin'** and **password = '1234'**, print Access Granted.

Else, Access Denied.

What I Did (Step by Step):

- Took username and password input from the user.
- Checked if the entered values matched the required credentials.
- Used an if statement to grant or deny access.
- Displayed the result based on the input match.

Code Screenshots



Output Screenshots

```
PROBLEMS   OUTPUT   TERMINAL   PORTS
TERMINAL                                                    Python
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/login system.py"
Enter username: admin
Enter password: 1234
Access Granted
PS D:\MA_week 03 tasks>
Ln 12, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1
```

Learning and Challenges

- Learned how to compare multiple conditions using and.
- Practiced basic input validation and string comparison.
- Faced no major issues due to the simple structure.
- Understood basic logic used in authentication systems.

Task 05:

Ask user for attendance (%) and final marks.

If attendance ≥ 75 and marks ≥ 50 → Promote

Else → Not promoted.

What I Did (Step by Step):

- Took attendance percentage and final marks as input.
- Used if condition with and to check promotion eligibility.
- Displayed result based on the input values.
- Handled invalid input using try-except.

Code Screenshots

```python
# Task: Check Promotion Eligibility

try:

    Attendance = float(input("Enter your attendance percentage %: "))
    FinalMarks = float(input("Enter your final marks: "))

    if Attendance >= 75 and FinalMarks >= 50:
        print("Promote")

    else:
        print("Not promoted")

except ValueError:
    print("Error: Please enter valid numeric values.")
```

Output Screenshots

```
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/Attendence and marks finder.py"
Enter your attendance percentage %: 96
Enter your final marks: 80
Promote
PS D:\MA_week 03 tasks>
```

Learning and Challenges

- Learned how to apply multiple conditions using logical and.
- Practiced input handling and numeric comparison.
- Ensured correct logic flow for real-life decision-making.
- Resolved input type issues using float() and error handling.

Task 06:

Billing system:

Take number of products and total price.

If price > 1000 and products > 3 → 15% discount

If price > 500 → 10% discount

Else → No discount.

Show final bill.

What I Did (Step by Step):

- Took user input for product count and total price.
- Used conditions to apply the correct discount rule.
- Calculated and displayed the discount and final bill.
- Handled invalid input types safely using try-except.

Code Screenshots

```python
# Task: Billing System with Discount

try:

    NumberOfProducts = int(input("Enter the number of products: "))
    TotalPrice = float(input("Enter the total price: "))
    Discount = 0

    if TotalPrice > 1000 and NumberOfProducts > 3:
        Discount = TotalPrice * 0.15

    elif TotalPrice > 500:
        Discount = TotalPrice * 0.10

    else:
        Discount = 0

    FinalBill = TotalPrice - Discount

    print("Total Price:", TotalPrice)
    print("Discount Applied:", Discount)
    print("Final Bill to Pay:", FinalBill)

except ValueError:
    print("Error: Please enter valid numeric values.")
```

Output Screenshots

```
PS D:\MA_week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/MA_week 03 tasks/Billing System.py"
Enter the number of products: 5
Enter the total price: 2000
Total Price: 2000.0
Discount Applied: 300.0
Final Bill to Pay: 1700.0
PS D:\MA_week 03 tasks>
```

Learning and Challenges

. Learned how to calculate and apply percentage-based discounts.

. Practiced combining multiple conditions using and and elif.

. Faced input conversion issues and fixed using proper data types.

. Applied real-world billing logic to make the program functional.