TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Hamza Badshah

Intern ID: TN/IN01/PY/001

Email ID : hamzabadshah2592@gmail.com

Internship Domain : Python Development

Task Week : 03

Instructor Name : Mr Hassan Ali

Task 01 :

Description:

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

Perform the operation and display the result.

Handle division by zero safely.

What I Did:

- 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 Snippets

```python
print("Welcome to the Simple Python Calculator")

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

    if operator == '+':
        result = first_number + second_number

    elif operator == '-':
        result = first_number - second_number

    elif operator == '*':
        result = first_number * second_number

    elif operator == '/':
        if second_number != 0:
            result = first_number / second_number

        else:
            result = "Error: Cannot divide by zero."

    elif operator == '%':
        if second_number != 0:
            result = first_number % second_number

        else:
            result = "Error: Cannot perform modulus by zero."
```

File Edit Selection View Go Run ... Week 03 tasks

EXPLORER

WEEK 03 TASKS

Simple Python Operation Calculator....

Welcome | Simple Python Operation Calculator.py

Simple Python Operation Calculator.py > ...

```python
        else:
            result = "Error: Cannot perform modulus by zero."

    elif operator == '//':
        if second_number != 0:
            result = first_number // second_number

        else:
            result = "Error: Cannot perform floor division by zero."

    elif operator == '**':
        result = first_number ** second_number

    else:
        result = "Invalid operator entered."

    print("Result:", result)

except ValueError:
    print("Invalid input. Please enter numeric values only.")
```

OUTLINE

TIMELINE

Output Snippets

```
PROBLEMS    OUTPUT    TERMINAL    PORTS
TERMINAL                                                    Python

PS D:\Week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/Week 03 tasks/Simple Python Operation Calculator.py"
Welcome to the Simple Python Calculator
Enter the first number: 3
Enter an operator (+, -, *, /, %, //, **): *
Enter the second number: 3
Result: 9.0
PS D:\Week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/Week 03 tasks/Simple Python Operation Calculator.py"
Welcome to the Simple Python Calculator
Enter the first number: 5
```

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:

Description:

Take marks of 3 subjects.

Calculate total, percentage and assign grade:

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

What I Did:

- 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 Snippets

```python
print("Grade Calculator")

try:
    subject_1 = float(input("Enter marks for Subject 1: "))
    subject_2 = float(input("Enter marks for Subject 2: "))
    subject_3 = float(input("Enter marks for Subject 3: "))

    total_obtained_marks = subject_1 + subject_2 + subject_3
    percentage = (total_obtained_marks / 300) * 100

    if percentage >= 85:
        grade = "A"

    elif percentage >= 70:
        grade = "B"

    elif percentage >= 50:
        grade = "C"

    else:
        grade = "Fail"

    print("\n--- Result ---")
    print("Total Obtained Marks:", total_obtained_marks)
    print("Percentage:", round(percentage, 2), "%")
    print("Grade:", grade)

except ValueError:
    print("Invalid input. Please enter numeric values only.")
```

Output Snippets

```
TERMINAL                                                    Python
Enter marks for Subject 1: 89
Enter marks for Subject 2: 65
Enter marks for Subject 3: 45

--- Result ---
Total Obtained Marks: 199.0
Percentage: 66.33 %
Grade: C
PS D:\Week 03 tasks>
```

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:

Description:

Ask user for monthly income and expenses.

Calculate savings and classify:

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

What I Did:

- 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 Snippets

```python
print("Saving and Expenses Calculator")

try:
    monthly_income = float(input("Enter your monthly income: "))
    monthly_expenses = float(input("Enter your monthly expenses: "))

    savings = monthly_income - monthly_expenses

    print(f"\nYour savings this month are {savings}")

    if savings > 10000:
        status = "Saving Well"

    elif 5000 <= savings <= 9999:
        status = "Average"

    else:
        status = "Try to Save"

    print(f"Status {status}")

except ValueError:
    print("Invalid input! Please enter numeric values only.")
```

Output Snippets

```
PS D:\Week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/Week 03 tasks/Saving and Expenses Calculator.py"
Saving and Expenses Calculator
Enter your monthly income: 700000
Enter your monthly expenses: 6543

Your savings this month are 693457.0
Status Saving Well
PS D:\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:

Description:

Build a login system. Ask username & password.

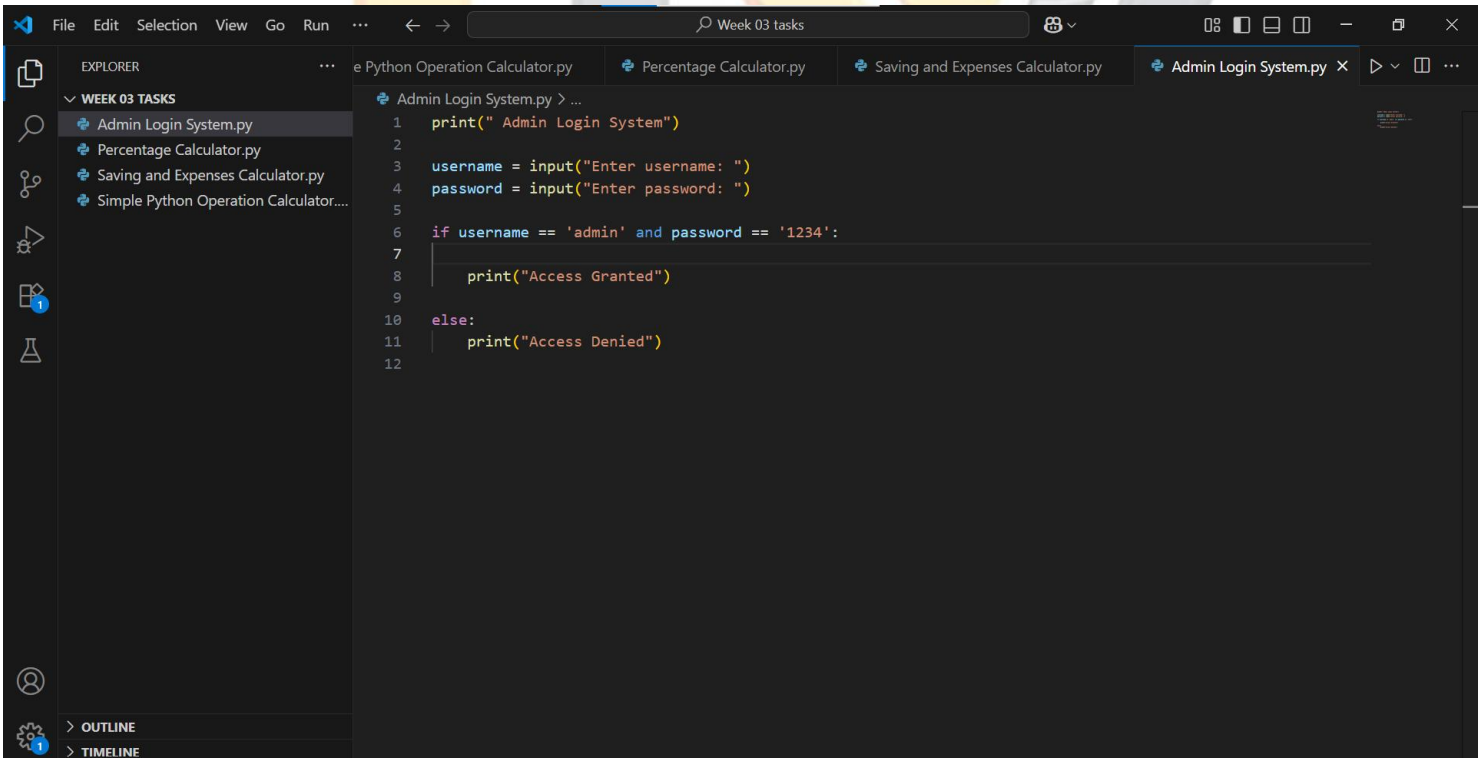
If username = 'admin' and password = '1234', print Access Granted.

Else, Access Denied.

What I Did:

- 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 Snippets



Output Snippets

```
PROBLEMS   OUTPUT   TERMINAL   PORTS
TERMINAL                                                    Python
PS D:\Week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/Week 03 tasks/Admin Login System.py"
 Admin Login System
Enter username: admin
Enter password: 1234
Access Granted
PS D:\Week 03 tasks>
```

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:

Description:

Ask user for attendance (%) and final marks.

If attendance ≥ 75 and marks ≥ 50 →

Promote Else → Not promoted.

What I Did:

- 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 Snippets

```python
print("Promotion Eligibility Checker")

try:
    attendance = float(input("Enter attendance percentage: "))
    final_marks = float(input("Enter final marks: "))

    if attendance >= 75 and final_marks >= 50:
        print("Status: Promoted")

    else:
        print("Status: Not Promoted")

except ValueError:
    print("Invalid input. Please enter numeric values only.")
```

Output Snippets

```
PS D:\Week 03 tasks> & "C:/Program Files/Python313/python.exe" "d:/Week 03 tasks/Grade Promoter Checker.py"
Promotion Eligibility Checker
Enter attendance percentage: 78
Enter final marks: 89
Status: Promoted
PS D:\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:

Description:

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:

- 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 Snippets

File Edit Selection View Go Run ... Week 03 tasks

EXPLORER

WEEK 03 TASKS

- Admin Login System.py
- Automatic Billing System.py
- Grade Promoter Checker.py
- Percentage Calculator.py
- Saving and Expenses Calculator.py
- Simple Python Operation Calculator....

OUTLINE

TIMELINE

Saving and Expenses Calculator.py | Admin Login System.py | Grade Promoter Checker.py | Automatic Billing System.py

Automatic Billing System.py > ...

```python
print("Billing System with Discount")

try:

    number_of_products = int(input("Enter number of products: "))
    total_price = float(input("Enter total price: "))

    if total_price > 1000 and number_of_products > 3:
        discount = 0.15 * total_price

    elif total_price > 500:
        discount = 0.10 * total_price

    else:
        discount = 0.0

    final_amount = total_price - discount

    print("\n--- Final Bill ---")
    print(f"Original Price {total_price}")
    print(f"Discount Applied {round(discount,2)}")
    print(f"Final Amount to Pay {round(final_amount,2)}")

except ValueError:

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

Output Snippets

```
PROBLEMS    OUTPUT    TERMINAL    PORTS
TERMINAL                                                    Python

Billing System with Discount
Enter number of products: 5
Enter total price: 10000

--- Final Bill ---
Original Price 10000.0
Discount Applied 1500.0
Final Amount to Pay 8500.0
PS D:\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.

Conclusion:

Each task helped me apply programming logic to practical problems such as finance tracking, grading systems, and user authentication. Challenges like input validation, division by zero, and combining multiple conditions enhanced my debugging skills and taught me the importance of clean, readable code.

Overall, this week's tasks helped me build confidence in Python basics and prepared me for more complex programming scenarios ahead. I now feel more comfortable writing structured programs and solving real-life problems through code.