Jan 28 Control Structures, IO, Loops, Functions

January 27, 2025 5:30 PM

Q1. Write a program that:

a. Prompts the user to enter their name.
b. Displays a message saying, **"Welcome to PASS, [NAME]"**, where [NAME] is the name entered by the user.

```
Enter your name: Jasan
Welcome to PASS, Jasan!
```

Q2. Next, modify the program to include a Boolean variable named is_new_student. Set is_new_student to False. Based on the value of this variable:

- If is_new_student is False, the program should display: **"Welcome Back to PASS, [NAME]"**.
- If is_new_student is True, the program should display: **"Welcome to PASS, [NAME]"**.

```
Enter your name: Jasan
Welcome Back to PASS, Jasan!
```

Q3. How will you change in your current program so that the program displays "**Welcome to PASS, [NAME]**"

Q4. Update the program to do the following:

a. Ask the user to enter a name **five times** (one name at a time).
b. For each name entered, the program should display a welcome message.
    - If the is_new_student variable is True, the message should be: **"Welcome to PASS, [NAME]"**.
    - If the is_new_student variable is False, the message should be: **"Welcome Back to PASS, [NAME]"**.

The program should repeat the process for all five names and display a separate welcome message for each name entered.

```
Enter name: Jasan
Welcome Back to PASS, Jasan!
Enter name: Ruby
Welcome Back to PASS, Ruby!
Enter name: Francis
Welcome Back to PASS, Francis!
Enter name: Pujan
Welcome Back to PASS, Pujan!
Enter name: Simran
Welcome Back to PASS, Simran!
```

Q5. Rewrite the program so that:

a. The entire program is enclosed within a function called welcome_students.
b. The function should take one parameter, is_new_student, which is a Boolean value.

- If is_new_student is True, the program should display: **"Welcome to PASS, [NAME]"**.
- If is_new_student is False, the program should display: **"Welcome Back to PASS, [NAME]"**.

c. The function should:
- Ask the user to enter a name **five times** (one name at a time).
- Display the appropriate welcome message for each name entered, based on the value of is_new_student.

d. Call the function by passing True or False as the argument for is_new_student.

Q6. Write a Python function named describe_weather that:

a. Takes one parameter:
- temperature (an integer representing the temperature in degrees Celsius).

b. Based on the value of temperature, the function should display the following weather descriptions:
- If temperature is less than -15, display: **"It's extremely cold! Frostbite is a serious risk. Stay indoors and dress in layers if you must go out."**.
- If temperature is between -15 and -1 (inclusive), display: **"It's freezing cold! Stay indoors and keep warm."**.
- If temperature is between 0 and 5 (inclusive), display: **"It's very cold outside. Wear a heavy jacket."**.
- If temperature is between 6 and 10 (inclusive), display: **"It's cold. Make sure to dress warmly."**.
- If temperature is between 11 and 15 (inclusive), display: **"It's chilly. A light jacket should be enough."**.
- If temperature is between 16 and 25 (inclusive), display: **"The weather is pleasant."**.
- If temperature is greater than 25, display: **"It's hot outside!"**.

c. Call the function with different temperature values to test that it displays the correct description.

Q7. Write a Python Function to Find the Maximum of Two Numbers

a. Function Name:
Create a function named max_of_two.

b. Parameters:
The function should take two parameters:
- num1 (the first number).
- num2 (the second number).

c. Functionality:
The function should:
- Compare the two numbers.
- **Return** the larger of the two numbers.

d. Example:
If the function is called as:

```
result = max_of_two(5, 8)
print(result)
```

Q8. Write a Python Function to Find the Maximum of Three Numbers Using Nested Conditional Statements

1. Function Name:
Create a function named max_of_three.

2. **Parameters**:
   The function should take three parameters:
   1. a (the first number).
   2. b (the second number).
   3. c (the third number).
3. **Functionality**:
   The function should:
   1. Use **nested conditional statements** (if, else) to compare the three numbers.
   2. Determine and return the largest of the three numbers.
4. **Output**:
   When you call the function, it should return the maximum number from the three inputs.
5. **Example**:
   If the function is implemented as follows:

```
print(max_of_three(2, 4, 6))  # Test case 1
print(max_of_three(6, 4, 2))  # Test case 2
print(max_of_three(2, 6, 4))  # Test case 3
print(max_of_three(4, 2, 6))  # Test case 4
print(max_of_three(6, 2, 4))  # Test case 5
print(max_of_three(4, 6, 2))  # Test case 6
```

   The expected output is:

```
6
6
6
6
6
6
```

6. **Hint**:
   1. Use an outer if-else statement to compare the first two numbers (a and b).
   2. Inside each block, use another if-else statement to compare the result with the third number (c).

Q9. Modify the Function to Find the Maximum of Three Numbers Using Logical Operators

1. **Objective**:
   Update the max_of_three function to use **logical operators** (like and) instead of nested conditional statements.
2. **Function Name**:
   The function should still be named max_of_three.
3. **Parameters**:
   The function should take three parameters:
   - a (the first number).
   - b (the second number).
   - c (the third number).
4. **Functionality**:
   The function should:
   - Use logical operators (and) to check multiple conditions in a single if statement.
   - Determine the largest number among a, b, and c.
   - Return the largest number.