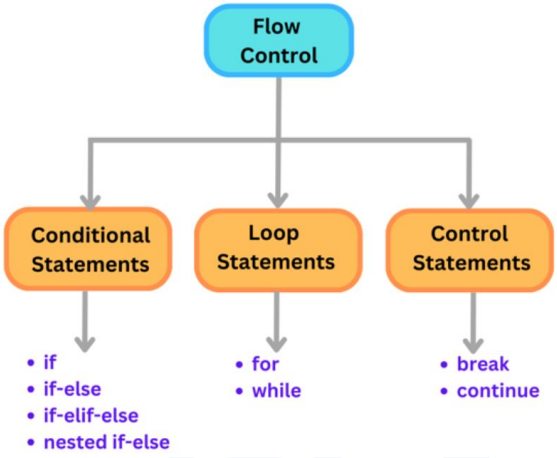
Lesson Plan:

Conditions (If Else, If-Elif-Else)

PW SKILLS

Topics to be covered:

1. Conditional Statements
2. If statements
3. If-else statements
4. if-elif-else statements
5. Nested if-else statements
6. Analogy of Conditional Statements



1. Conditional Statements:

- Conditional statements, often known as control structures, are an essential component of programming. They enable you to make code decisions based on predefined conditions.

- The primary conditional statements in Python are if, elif (short for "else if"), and else.

Purpose of Conditional Statements:

The purpose of conditional statements in programming is to enable the execution of different code blocks based on specific conditions. Here are four key purposes of conditional statements:

1. Decision Making: Conditional statements allow a program to make decisions by evaluating conditions. They help determine which set of instructions to execute depending on whether the conditions are met or not.

2. Control Flow: Conditional statements control the flow of a program, ensuring that the code follows a specific path or branches based on the given conditions. This helps create responsive and adaptable programs.

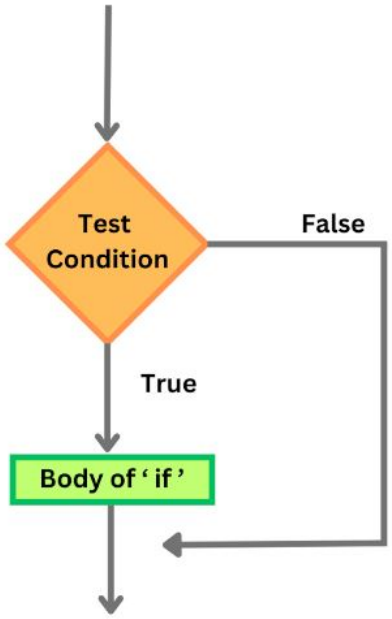
3. Error Handling: Conditional statements are used for error handling. They can identify and respond to exceptional situations, preventing program crashes or unexpected behavior by executing specific error-handling code.

4. Customization: Conditional statements provide the means to customize the behavior of a program for different scenarios or inputs. They allow programmers to tailor the program's response to various user interactions or data inputs, making the software more versatile and user-friendly.

2. 'if' statements:

- The if statement is the most basic type of control statement. It accepts a condition and determines whether it is True or False.

- If the condition is True, the True piece of code is run; otherwise, the block of code is bypassed, and the controller proceeds to the next line.



Examples:

```
x = 10

if x > 5:
    print("x is greater than 5")
```

```
temperature = 30

if temperature > 25:
    print("It's a hot day")
```

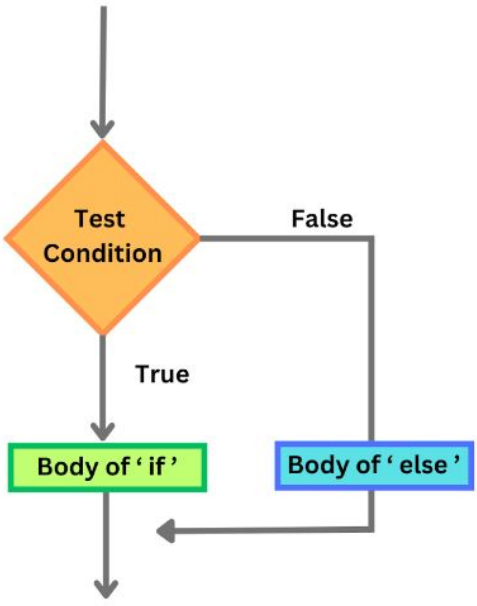
```
is_datascience_course = True

if is_datascience_course:
    print("Don't forget to join PWskills")
```

```
age = 18
if age ⩾ 18:
    print("You are an adult.")
else:
    print("You are not an adult.")
```

```
score = 75
passing_score = 70
if score ⩾ passing_score:
    print("Congratulations, you passed!")
else:
    if score ⩾ passing_score - 5:
        print("You almost passed.")
    else:
        print("You didn't pass.")
```

3. 'if-else' statements:

- The if-else statement checks the condition and executes the if block of code when the condition is True, and if the condition is False, it will execute the else block of code.



Examples:

```
x = 10
if x > 5:
    print("x is greater than 5")
else:
    print("x is not greater than 5")
```

```
age = 20
if age ⩾ 18:
    print("You are eligible to vote")
else:
    print("You are not eligible to vote")
```

```
is_datascience_course = True

if is_datascience_course:
    print("Don't forget to join PWskills")
else:
    print("ASAP this course will be there")
```

```
num = 7
if num % 2 == 0:
    print("Even")
else:
    print("Odd")
```

```
score = 85
result = "Pass" if score ⩾ 70 else "Fail"
print(f"You {result}.")
```

4. 'if-elif-else' statements:

- The if-elif-else condition statement in Python uses elif blocks to link multiple conditions one after the other. This is useful when you need to check numerous conditions at the same time.

- We can make a difficult decision with the help of if-elif-else. The elif statement checks each condition one by one and executes the code if the condition is met.

Examples:

```
x = 10
if x > 5:
    print("x is greater than 5")
elif x == 5:
    print("x is equal to 5")
```

```
score = 75
if score >= 90:
    print("A")
elif score >= 80:
    print("B")
elif score >= 70:
    print("C")
```

```
hour = 14

if hour < 12:
    print("Good morning")
elif hour < 17:
    print("Good afternoon")
else:
    print("Good evening")
```

```
num = 0
if num > 0:
    print("Positive")
elif num < 0:
    print("Negative")
else:
    print("Zero")
```

```
age = 30
if age < 18:
    print("You are a minor.")
elif 18 <= age < 65:
    print("You are an adult.")
else:
    print("You are a senior citizen.")
```

5. 'Nested if-else' statements:

- In Python, a nested 'if-else' statement is a construct in which one 'if' statement is contained within another 'if' or 'else' block.
- This enables you to design a condition hierarchy in which the inner 'if-else' statements are evaluated only if the outer condition is true.
- When you need to test many circumstances in an organized manner, nested if-else statements are useful.

Examples:

```
x = 10
y = 5

if x > 5:
    if y > 5:
        print("Both x and y are greater than 5.")
    else:
        print("x is greater than 5, but y is not.")
else:
    print("x is not greater than 5.")
```

```
is_weekend = False
is_sunny = True

if is_weekend:
    if is_sunny:
        print("Go for a picnic.")
    else:
        print("Stay in and relax.")
else:
    print("It's a workday.")
```

```
is_student = True
is_teacher = False

if is_student:
    if is_teacher:
        print("You are both a student and a teacher.")
    else:
        print("You are a student but not a teacher.")
else:
    if is_teacher:
```

```
        print("You are a teacher but not a student.")
    else:
        print("You are neither a student nor a teacher.")
```

```
is_vip = True
age = 30

if is_vip:
    if age >= 18:
        if age < 65:
            print("Welcome, VIP customer!")
        else:
            print("You're a VIP, but you qualify for senior
discounts.")
    else:
        print("VIP status is for adults only.")
else:
    print("Regular pricing applies.")
```

Analogy of Conditional Statements:

- Control statements, using `if`, `if-elif-else`, and `if-else`, can be compared to making decisions in everyday life:

Using `if` Statements:

- Imagine you're deciding whether to go outside to play. If the weather is sunny, you'll go outside; otherwise, you'll stay indoors.
    - If it's sunny (condition met), you go outside.
    - If it's not sunny (condition not met), you stay indoors.

Using `if-elif-else` Statements:

- Now, think of planning a day at an amusement park. You have different age groups with different ticket prices: children, adults, and seniors.
    - If you are a child, you pay the child's price.
    - If you are an adult, you pay the adult's price.
    - If you are a senior, you pay the senior's price.

Using `if-else` Statements:

- Consider a scenario where you're checking whether a fruit is ripe before eating it. If it's ripe, you eat it; otherwise, you leave it for later.
    - If the fruit is ripe (condition met), you eat it.
    - If the fruit is not ripe (condition not met), you don't eat it.
- In all these analogies, the `if` statements determine the course of action based on a single condition. The `if-elif-else` statements help you make choices from a range of options, and the `if-else` statements provide a simple binary choice.