4. Conditional Statements :

❖Introduction to conditional statements: if, else, elif.

Conditional statements allow your program to execute certain blocks of code only when specific conditions are met. They enable decision-making capabilities within your code.

if Statement:

The if statement evaluates a condition and executes the following indented block if the condition is True.

Syntax:

```
if condition:
    # Code to execute if condition is True
```

else Statement:

The else statement follows an if or elif and executes its block if none of the preceding conditions were True.

Syntax:

```
if condition:
    # Code if condition is True
else:
    # Code if condition is False
```

elif Statement:

Short for "else if," the elif statement allows you to check multiple conditions sequentially. It only evaluates its condition if all preceding if and elif conditions were False.

Syntax:

```
if condition1:
    # Code if condition1 is True
elif condition2:
    # Code if condition2 is True
else:
    # Code if all conditions are False
```

❖ Nested if-else conditions.

In Python, nested if-else statements allow you to perform multiple conditional checks within each other, enabling complex decision-making processes. However, it's important to use them judiciously, as excessive nesting can reduce code readability and maintainability.

Nested if-else:

A nested if-else statement involves placing an if or else block inside another if or else block:

Syntax:

```
if condition1:
    if condition2:
        # Execute if both condition1 and condition2 are true
    else:
        # Execute if condition1 is true and condition2 is false
else:
    # Execute if condition1 is false
```