***Quaid\_e\_Awam University of Engineering Science & Technology***

* ***Areeba Ateeque***
* ***23BSAI52***
* ***Programming Fundamental***
* ***PYTHON CONTROL FLOW (IF ELSE, ELIF, NESTED IF ELSE)***
* ***ASSIGNMENT NO # 11***

**--------------------------------------------------------------------------------------------------------------------**

**Qno1**:**What is the purpose of using control flow statement like if else and elif in python?**

**Ans**:Control flow statements like if, else, and elif in Python are used to control the execution flow of a program based on certain conditions. They allow you to make decisions and execute different blocks of code depending on whether a given condition is true or false.

•if statement: Executes a block of code if the specified condition is true.

•else statement: Provides an alternative block of code to be executed if the if condition is false.

•elif statement: Stands for “else if” and allows you to check multiple conditions in a sequential manner. If the previous if or elif condition is false, it moves on to the next one.

**Qno2:How does python determine which block of code to execute in an if else statement ?**

**Ans:** In Python, the if-else statement determines which block of code to execute based on the condition provided in the if clause. If the condition is true, the code inside the if block is executed; otherwise, the code inside the else block is executed. The condition is evaluated as either True or False, and the corresponding block is executed accordingly.

**Qno 3:Explain the difference between the if-elif-else and nested if-else structures?**

**Ans**:The if-elif-else structure allows you to check multiple conditions and execute different blocks of code based on the first condition that evaluates to True. It provides a way to handle multiple cases without having to write separate if statements for each case. The elif (short for “else if”) keyword is used to add additional conditions to check.

On the other hand, the nested if-else structure is when an if statement is placed inside another if statement. This allows for more complex conditional logic. The inner if-else statement is only executed if the outer if statement evaluates to True.

**\*Example\*:**

***If-elif-else structure:***

```python

X = 5

If x < 5:

Print(“x is less than 5”)

Elif x == 5:

Print(“x is equal to 5”)

Else:

Print(“x is greater than 5”)

***Nested if-else structure:***

```python

X = 5

If x < 10:

If x < 5:

Print(“x is less than 5”)

**Else**:

Print(“x is between 5 and 10”)

**Else**:

Print(“x is greater than or equal to 10”)

In the if-elif-else structure, only one block of code is executed based on the first condition that is true. In the nested if-else structure, the inner if-else statement is only executed if the outer if statement is true.

**Qno4**:**How can you use logical operators (and, or, not) with if-else statements in Python?**

**Ans:** we can use logical operators (and, or, not) with if-else statements in Python to create more complex conditions.

* The “and” operator is used to check if multiple conditions are true. If all conditions connected by “and” are true, the overall condition evaluates to true.
* The “or” operator is used to check if at least one of the conditions is true. If any of the conditions connected by “or” is true, the overall condition evaluates to true.
* The “not” operator is used to negate a condition. It returns the opposite of the condition’s result.

\***Example**\*

```python

X = 5

Y = 10

If x > 0 and y > 0:

Print(“Both x and y are positive.”)

If x > 0 or y > 0:

Print(“At least one of x or y is positive.”)

If not x > 10:

Print(“x is not greater than 10.”)

**Qno5:Describe scenarios where nested if-else statements are preferred over if-elif-else structures?**

**Ans:**Nested if-else statements are often preferred over if-elif-else structures in a few scenarios.

***Example***: when we have multiple conditions that need to be checked independently, rather than in a sequential manner. Another scenario is when we want to add more complexity to our code by nesting multiple if-else statements within each other. This can be useful when we have specific conditions that need to be met before moving on to the next set of conditions.

**Qno6:How does Python handle multiple conditions in an if-elif-else ladder?**

**Ans:**In Python, an if-elif-else ladder allows you to check multiple conditions in a sequential manner. The conditions are evaluated one by one, from top to bottom. When a condition evaluates to True, the corresponding block of code associated with that condition is executed, and the rest of the ladder is skipped. If none of the conditions are True, the code inside the else block, if present, is executed. This allows you to handle different cases based on the conditions in a structured way.

**Qno7:Why is it important to indent properly when using control flow statements in Python?**

**Ans:**Proper indentation is crucial when using control flow statements in Python because it determines the structure and execution flow of your code. Python uses indentation to define blocks of code, such as the body of an if statement or a loop. By indenting your code correctly, you clearly indicate which statements belong to a particular block and ensure that they are executed in the intended order. Incorrect indentation can lead to syntax errors or unexpected behavior in your program. So, always remember to indent properly to maintain the clarity and readability of your code.