WEEK-3
Control Flow: Conditional blocks If statement: general format; Multiway branching; Sufficient examples;

Control Flow Statements

- Python supports a set of control flow statements that you can integrate into your program.
- The statements inside your Python program are generally executed sequentially from top to bottom, in the order that they appear.
- Apart from sequential control flow statements you can employ decision making and looping control flow statements to break up the flow of execution thus enabling your program to conditionally execute particular blocks of code.

The control flow statements in Python Programming Language are

- Sequential Control Flow Statements: This refers to the line by line execution, in which the statements are executed sequentially, in the same order in which they appear in the program.
- Decision Control Flow Statements: Depending on whether a condition is True or False, the decision structure may skip the execution of an entire block of statements or even execute one block of statements instead of other (if, if…else and if…elif…else).
- Loop Control Flow Statements: This is a control structure that allows the execution of a block of statements multiple times until a loop termination conditionis met (for loop and while loop). Loop Control Flow Statements are also called Repetition statements or Iteration statements.

Control Flow: Conditional blocks

- In Python, conditional block statements execute depending on whether a given condition is True or False. We can execute different blocks of codes depending on the outcome of a condition.
- Condition statements always evaluate to either True or False.
- Python programming language assumes any non-zero and non-null values as TRUE, and any zero or null values as FALSE value.

if statement:

- It is a one way branching statement.
- The if statement checks the condition and executes the if block of code when the condition is True, and if the condition is False, it will not execute it.
- The syntax of if statement is given below

Syntax:

if(expression):

statement(s)

- If the condition is True, then statement(s) will be executed If the condition is False, statement(s) will not be executed.

Where,

- ☞ if is a keyword.
- ☞ Expression: valid Boolean expression results in either True or False
- ☞ Colon should be present at the end of if statement.
- ☞ If block statement should have proper indentation.

- Flowchart of if statement:

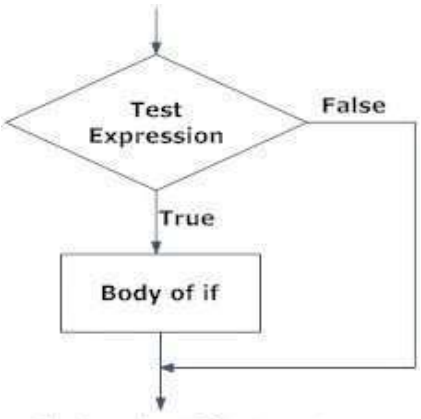

Fig: Operation of if statement

Example1:Program Reads a Number and Prints a Message If It Is

Positivenumber = int(input("Enter a number"))

if number >= 0:

print(f"The {number} entered by the user is a positive number")

if...else Statement

- An if statement can also be followed by an else statement which is optional.
- An else statement does not have any condition.
- Statements in the if block are executed if the Boolean_Expression is True.
- Use the optional else block to execute statements if the Boolean_Expression

isFalse.

- The if...else statement allows for a two-way decision.
- **The syntax is as follows**

```
if test expression:
   Body of if
else:
   Body of else
```

- The flowchart of if.. else is shown as follows:

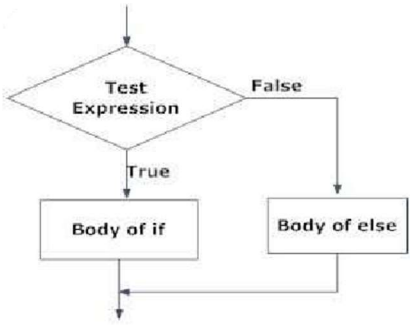

Fig: Operation of if...else statement

Example 1:Program to find the given number is odd or even

```
number = int(input("Enter a number"))
if number % 2 == 0:
        print(f"{number} is Even number")
else:
        print(f"{number} is Odd number")
```

Example 2: Program to find the given number is positive or negative

```
num = int(input("Enter a number"))
if num >= 0:
   print("Positive or Zero")
else:
   print("Negative number")
```

Example 3: Program to Find the Greater of Two Numbers

```
number_1 = int(input("Enter the first number"))
number_2 = int(input("Enter the second number"))if
number_1 > number_2:
        print(f"{number_1} is greater than {number_2}")
else:
      print(f"{number_2} is greater than {number_1}")
```

if…elif…else Statement

- The if…elif…else is also called as multi-way decision control statement.
- When you need to choose from several possible alternatives, then an elif statement is used along with an if statement.
- The else statement must always come last, and will again act as the default action.
- **Syntax:**

  if test expression:

  Body of if

  elif test expression:

  Body of elif

  else:

  Body of else

- The flowchart is shown the figure below

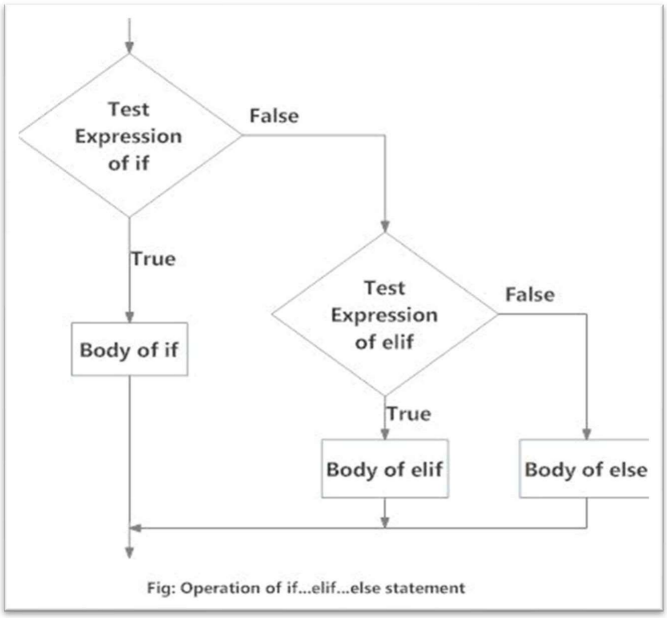

Fig: Operation of if...elif...else statement

Example 1:Program to find the given number is positive, zero or negative

```
num = int(input("Enter a number"))
if num > 0:
    print(f" {num} is Positive number")
```

```
elif num == 0:
    print("Zero")
else:
    print(f"{num} is Negative number")
```

Example 2: Program to Prompt for a Score between 0.0 and 1.0. If the Score Is Outof Range, Print an Error. If the Score Is between 0.0 and 1.0, Print a GradeUsing the Following Table

Score	Grade
>= 0.9	A
>= 0.8	B
>= 0.7	C
>= 0.6	D
< 0.6	F

```
score = float(input("Enter your score"))

if score < 0 or score > 1:
        print('Wrong Input')
elif score >= 0.9:
        print('Your Grade is "A" ')
elif score >= 0.8:
        print('Your Grade is "B" ')
elif score >= 0.7:
        print('Your Grade is "C" ')
elif score >= 0.6:
        print('Your Grade is "D" ')
else:
        print('Your Grade is "F" ')
```

Nested if Statement

- In some situations, you have to place an if statement inside another statement.
- An if statement that contains another if statement either in its if block or else block is called a Nested if statement.

- **The syntax of the nested if statement is**

```
if test_expression1:
    if test_expression2:
        statement1
    else:
        statement2

else:
    if test_expression3:
        statement3
    else:
        statement4
```

Example 1:Program to Check If a Given Year Is a Leap Year

```
year = int(input('Enter a year'))
if year % 4 == 0:
    if year % 100 == 0:
        if year % 400 == 0:
            print(f'{year} is a Leap Year')
        else:
            print(f'{year} is not a Leap Year')
    else:
        print(f'{year} is a Leap Year')
else:
        print(f'{year} is not a Leap Year')
```

statement2