Chapter 5 Conditional and Iterative Statements

Statement

Statement are the instructions given to the computer to perform any kind of action.

Types of Statement

1. Empty Statement

The statement which does nothing. In python an empty statement is pass statement. It takes the following form :

```
pass
```

wherever Python encounters a pass statement, Python does nothing and moves to next statement.

2. Simple Statement

Any single executable statement is a simple statement in Python. E.g.

```
print(“Programming”)
```

3. Compound Statement

A compound statement represents a group of statements executed as a unit. It takes following form :

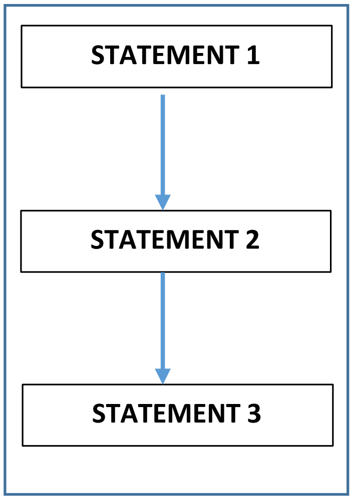
```
<compound statement header> :
  <indented body containing multiple simple/compound statements>
```

STATEMENT FLOW CONTROL

- In a program, statements may be executed sequentially, selectively or iteratively.
- Every programming language provides constructs to support sequence, selection or iteration.

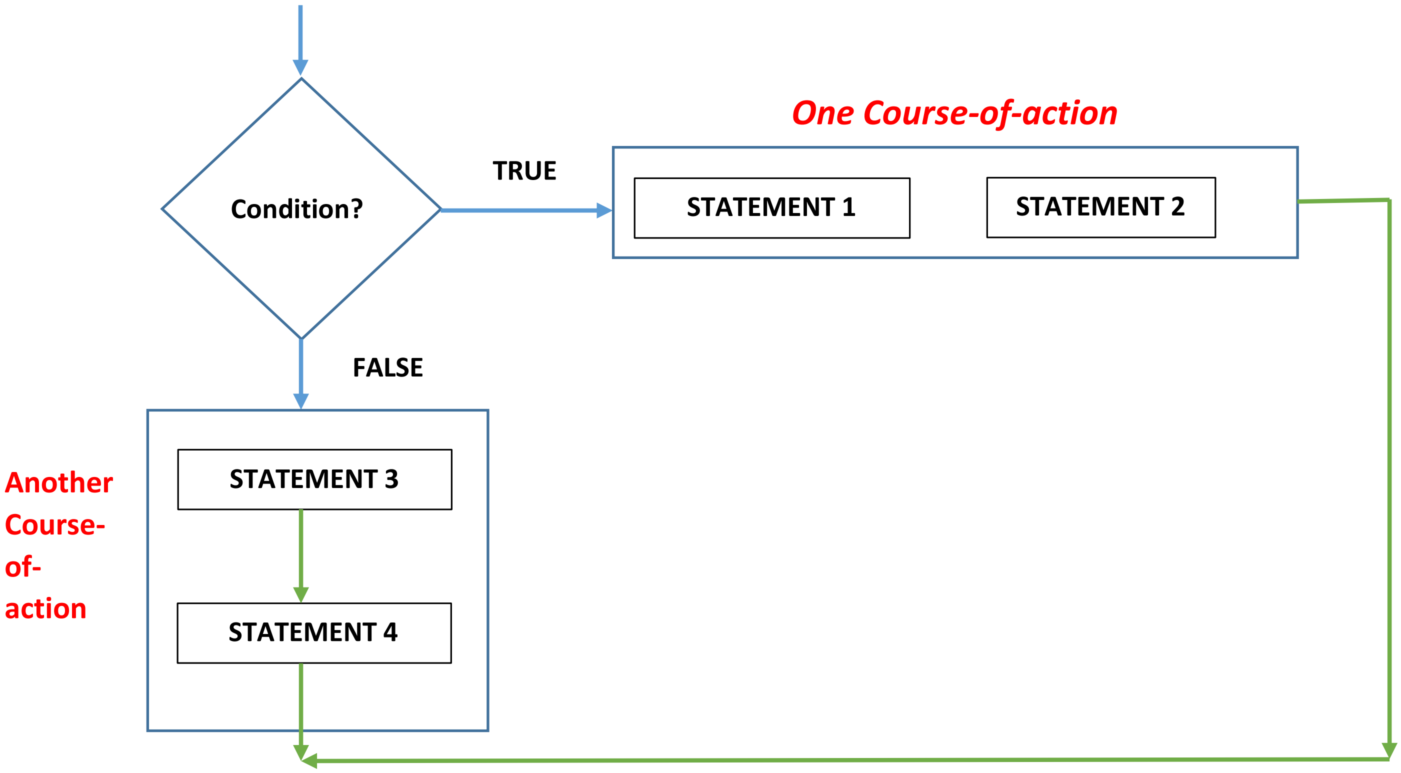
SEQUENCE

- The sequence construct means the statements are being executed sequentially.
- Java execution starts with first statement, and then each statement is executed in turn.



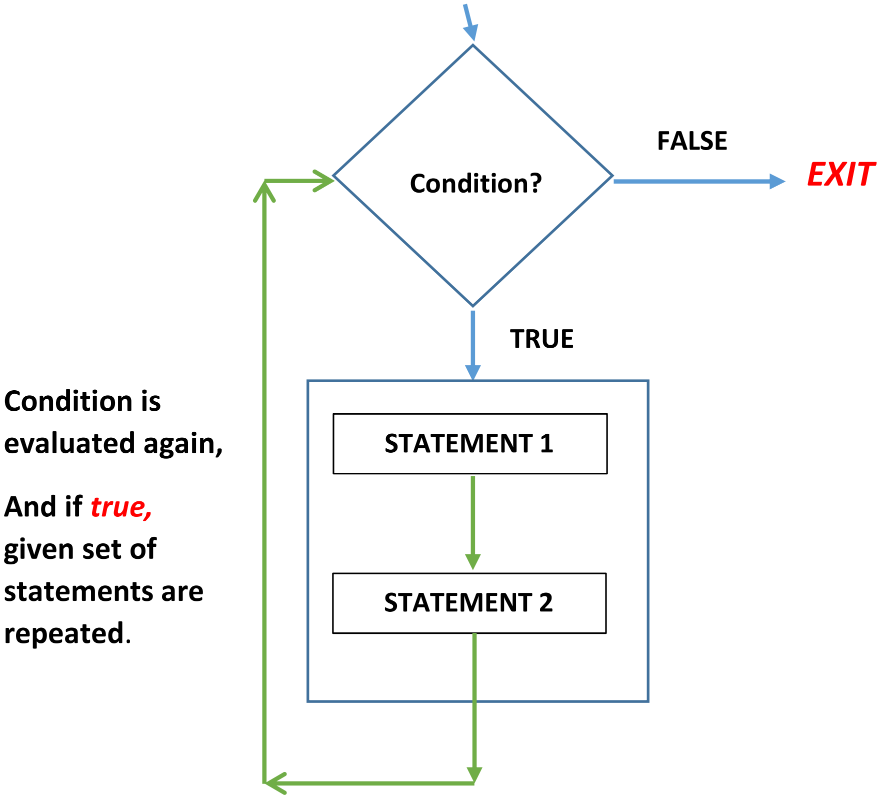
SELECTION

- The selection construct means the execution of statement(s) depending upon a condition-test.
- If a condition evaluates to true, a course of action (a set of statements) is followed otherwise another course of action (a different set of statements) is followed.



ITERATION

- The iteration construct means **repetition** of a set of statements depending upon a condition-test. Till the time a condition is true, a set of statements are repeated again and again. As soon as the condition becomes false, the repetition stops.
- The iteration construct is also called **looping construct**.


Condition?
FALSE
EXIT
TRUE
STATEMENT 1
STATEMENT 2
Condition is evaluated again,
And if true, given set of statements are repeated.

- The set of statements that are repeated again and again is called the **body of loop**.

PROGRAM LOGIC DEVELOPMENT TOOLS

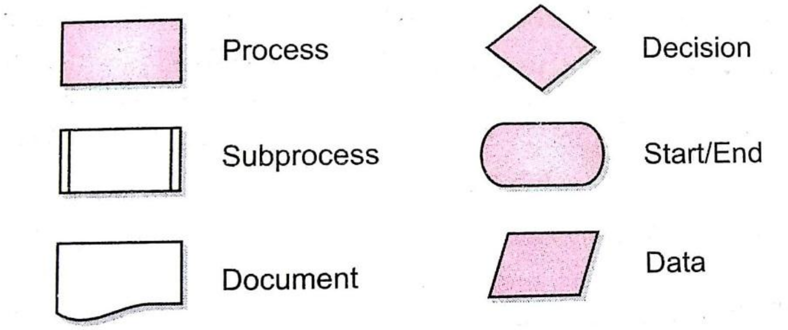
- Helps in reading and analysing the problem which needs to be solved by programming.

ALGORITHM

- It is a step by step procedure (well defined instructions) to solve a given problem. For instance, the algorithm for addition of two numbers is :
    1. Input ***first number***.
    2. Input ***second number***.
    3. Add ***first number*** with ***second number*** and store the result in third variable.
    4. Display the result.

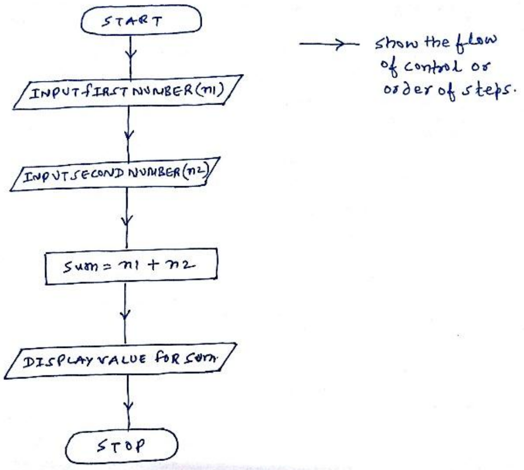
FLOWCHARTS

- It is a ***graphical representation of an algorithm***. Following symbols are used in flowcharts:



- Use **Data symbol** for Input/Output Operation (taking input and showing output)
- Use **Process symbol** for any type of computation and internal operations like initialization, calculation etc.

- e.g. flowchart for addition of two numbers will be :



PSEUDOCODE

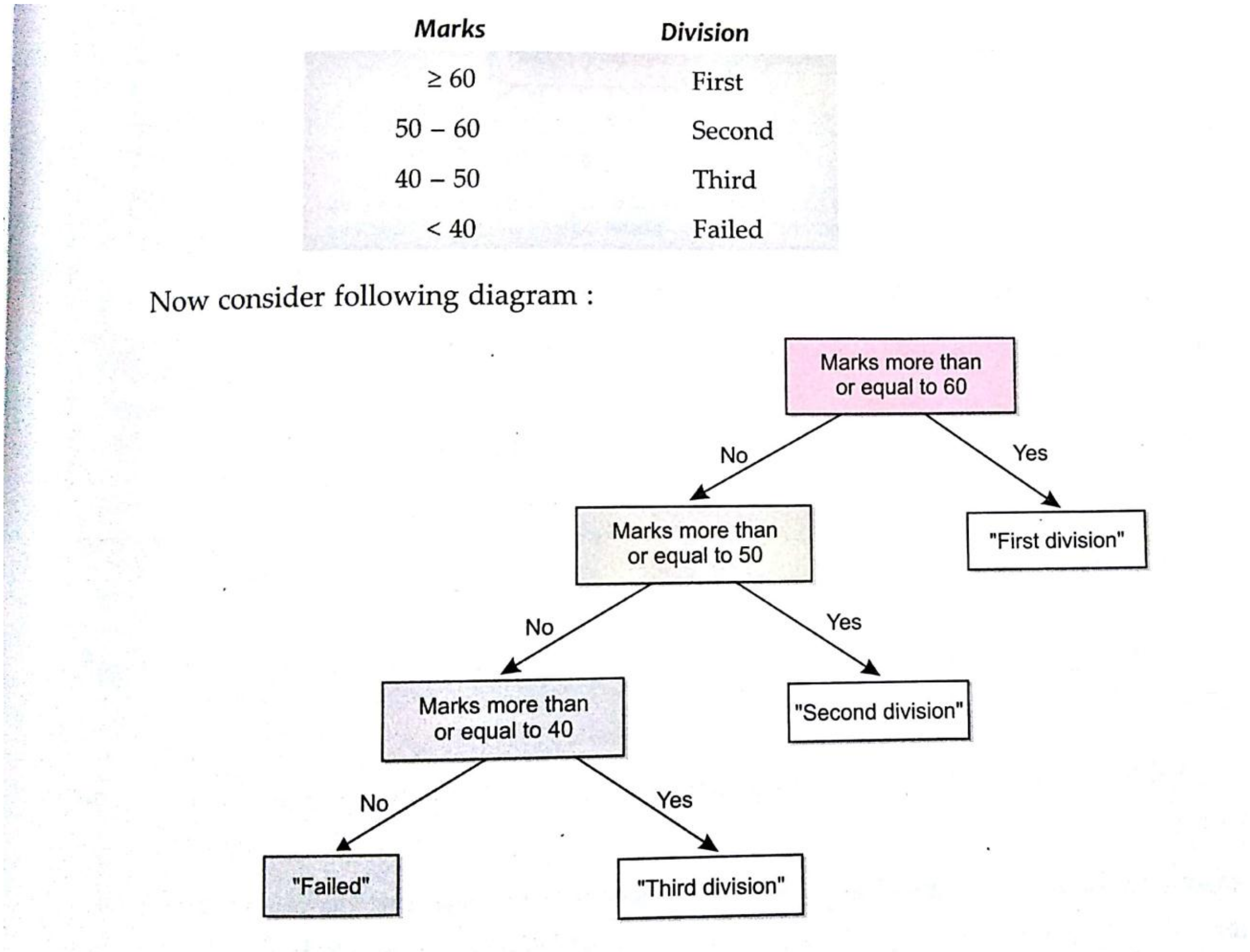
- It is an informal language that helps programmers describe steps of a program's solution without using any programming language syntax.

e.g. Pseudo code for addition of two number is :

```
Input first number in variable a
Input second number in variable b
Sum = firstnumber to be added with second number
Display the result
```

DECISION TREES

- These are a way of presenting rules in a hierarchical and sequential structure, where based on the hierarchy of rules, certain outcomes are predicted.



Marks	*Division*
≥ 60	First
50 – 60	Second
40 – 50	Third
< 40	Failed

Now consider following diagram :

Marks more than or equal to 60
No
Yes
Marks more than or equal to 50
"First division"
No
Yes
Marks more than or equal to 40
"Second division"
No
Yes
"Failed"
"Third division"

THE if STATEMENTS OF PYTHON

The if statements are the conditional statements in Python and these implement **selection constructs**.

The if statement

- if statement tests a **condition** and if the condition evaluates to **true**, it carries out some instructions and does nothing in case condition evaluates to **false**.

- if statement is a compound statement and its syntax (general form) is :

```
if <conditional/realtional expression> :
        statement
        [statement]
```

where a statement may consist of a single statement, a compound statement or just the **pass** statement.

e.g.1.

```
if ch == ‘ ‘ :
    print(“character is space”)
```

The above code will check whether the character variable ch stores a space or not ; if it does, then condition ch == ‘ ‘ evaluates to true and print statement given in second line will be executed.
If however, variable ch does not store a space i.e., the condition ch == ‘ ‘ evaluates to false, then nothing will happen, no statement will be executed.

e.g.2.

```
ch = input(“Enter a single character:”)
if ch >= ‘0’ and ch < ‘9’ :
        print(“You entered a digit”)
```

The above code after getting input in ch, compares its value; if the value falls between characters ‘0’ to ‘9’ i.e. the condition evaluates to true , and thus execute the statement in the if – body that is , it will print a message saying ‘You entered a digit’

e.g.3.

```
if grade == ‘A’ :
    print(“You did well”)
```

e.g.4. Program to find the eligibility of a person for voting. Display message “Eligible” or “Not Eligible for voting” based on age entered by the user. (using if statement)

```
age = float(input(“Enter your age:”))
if age >= 18 :
      print(“Eligible for voting”)
if age <= 18 :
      print(“Not Eligible for voting”)
```

e.g.5. Program that takes a number and checks whether the given number is odd or even.

```
Num = int(input(“Enter an integer:”))
if num % 2 == 0 :
   print(num, “is EVEN number”)
if num % 2 != 0 :
   print(num, “is ODD number”)
```

<u>The if-else statement</u>

- This form of if statement tests a condition and if the condition evaluates to true , it carries out statements indented below if and in case condition evaluates to false, it carries out statements indented below else.

- The Syntax(general form) of the if-else statement is as shown below :

```
if <conditional expression> :
          statement
          [statements]
else :
          statement
          [statements]
```

Following figure (Fig. 4.5) illustrates *if* and *if-else* constructs of Python.

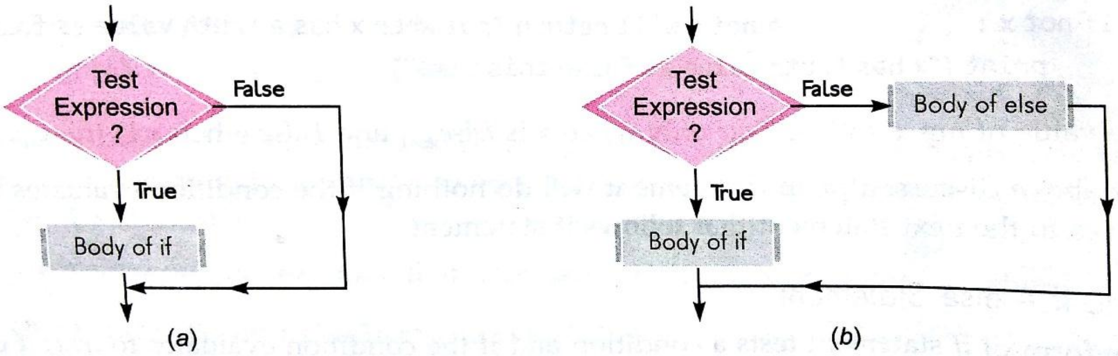


Figure 4.5 (*a*) The **if** statement's operation (*b*) The **if-else** statement's operation.

e.g.1.

```
if a > 0 :
     print(a, “is a positive number”)
else :
     print(b,”is a negative number”)
```

e.g.2. Program to accept two numbers and print the largest of two numbers.

```
x = float(input(“Enter first number:”))
y = float(input(“Enter second number:”))
if x > y :
    print(“largest number is”,x)
else :
    print(“largest number is”,y)
```

e.g.3. Program to find the eligibility of a person for voting. Display message “Eligible” or “Not Eligible for voting” based on age entered by the user.

```
age = float(input(“Enter your age:”))
if age >= 18 :
     print(“Eligible for voting”)
else :
     print(“Not Eligible for voting”)
```

e.g.4. Program that takes a number and checks whether the given number is odd or even.

```
num = int(input(“Enter an integer:”))
if num % 2 == 0 :
   print(num, “is EVEN number”)
else :
   print(num, “is ODD number”)
```

e.g.5. Program to test the divisibility of a number with another number.

```
num1 = int(input(“Enter first number:”))
num2 = int(input(“Enter second number:”))
rem = num1 % num2
if rem == 0 :
    print(num1,”is divisible by”, number2)
else :
    print(num1,”is not divisible by”, number2)
```

e.g.6. Program to display a menu for calculating area of circle or perimeter of a circle.

```
radius = float(input(“Enter radius of the circle:”))
print(“1. Calculate Area”)
print(“2. Calculate Perimeter”)
choice = int(input(“Enter your choice(1 or 2):”))
if choice == 1:
       area = 3.14 * radius * radius
       print(“Area of circle with radius”, radius, “is”, area)
else
       perm = 2 * 3.14159 * radius
       print(“Perimeter of circle with radius”, radius, “is”, perm)
```

The if – elif Statement

The general form of this statement is :

```
if <conditional expression> :
        statement
        [statements]
elif <conditional expression> :
        statement
        [statements]
else :
       statement
        [statements]
```

e.g. 1.

```
 if runs >= 100 :
        print(“Batsman scored a century”)
elif runs >= 50 :
        print(“Batsman scored a fifty”)
else :
        print(“Batsman has neither scored a century nor a fifty”)
```

Python will test this condition in case previous condition (runs >=100) is false

This block will be executed when both the if condition (i.e. runs >=100 and runs >=50) are false.

e.g. 2. Program to accept a number from user and check whether the number is positive,negative or zero.

```
num = float(input(“Enter a number:”))
 if num < 0 :
    print(num , “is a negative number”)
 elif num > 0 :
    print(num , “is a positive number”)
 else :
    print(num , “is equal to zero”)
```

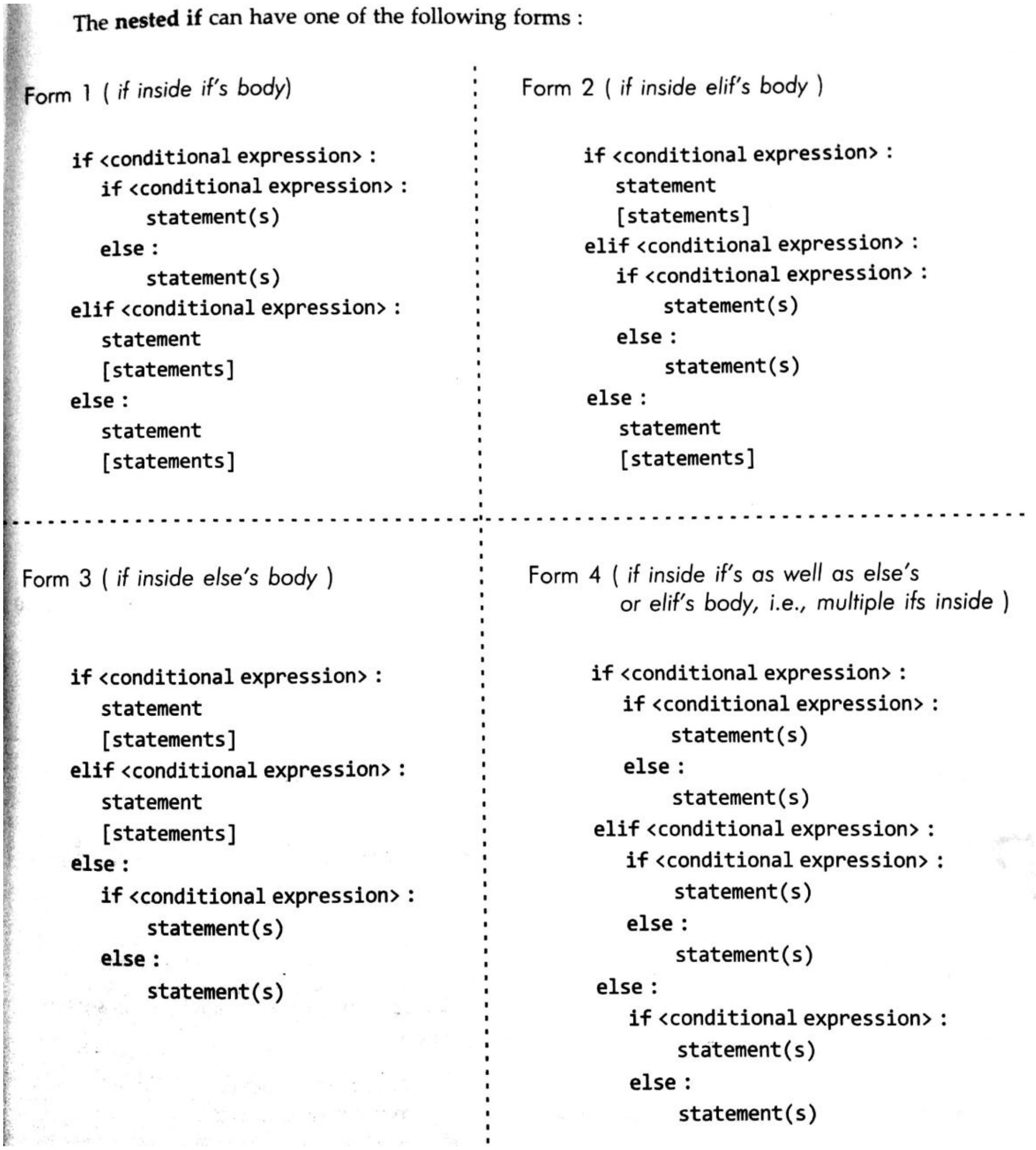
e.g. 3. Program that reads two numbers and an arithmetic operator and displays the computed result:

```
num1 = float(input(“Enter first number:”))
num2 = float(input(“Enter second number:”))
op = input(“Enter operator[+ , - , * , / , %]:”)
result = 0
if op == ‘+’ :
            result = num1 + num2
elif op == ‘-‘ :
            result = num1 – num2
elif op == ‘*’ :
```

```
                result = num1 * num2
        elif op == '/' :
                result = num1 / num2
         elif op == '%' :
                result = num1 % num2
         else :
                print("Invalid operator!!")
        print(num1,op,num2, '=" , result)
```

Nested if Statement

- A Nested if is an if that has another if in its if's body or in elif's body or in its else's body.



The **nested if** can have one of the following forms :

Form 1 (*if inside if's body*)

```
if <conditional expression> :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
elif <conditional expression> :
   statement
   [statements]
else :
   statement
   [statements]
```

Form 2 (*if inside elif's body*)

```
if <conditional expression> :
   statement
   [statements]
elif <conditional expression> :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
else :
   statement
   [statements]
```

Form 3 (*if inside else's body*)

```
if <conditional expression> :
   statement
   [statements]
elif <conditional expression> :
   statement
   [statements]
else :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
```

Form 4 (*if inside if's as well as else's or elif's body, i.e., multiple ifs inside*)

```
if <conditional expression> :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
elif <conditional expression> :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
else :
   if <conditional expression> :
       statement(s)
   else :
       statement(s)
```

STORING CONDITIONS

To make program readable, conditions in the if statement can be stored in name and then can be used in the if statements.

e.g.

```
pos = a > 0
neg = a < 0
num = float(input("Enter a number:"))
        if neg :
           print(num , "is a negative number")
        elif  pos :
           print(num , "is a positive number")
        else :
           print(num , "is equal to zero")
```

Practice Questions:

Q1. Correct the following code fragment:

```
if(x == 1)
    k = 100
else
     k = 10
```

Q2. What will be the output of following code fragment if the input given is (i) 7 and (ii) 5 ?

```
a = input("Enter a number:")
if (a == 5) :
  print("Five")
else :
   print("Not Five")
```

Q3. What is the output of following code?

```
if ( 4 + 5 == 10) :
    print("TRUE")
else :
    print("FALSE")
print("TRUE")
```

Q4. What is the output produced by the following code?

```
x =1
if x > 3 :
   if x > 4 :
       print("A", end= ' ')
    else :
       print("B",end= ' ')
elif x < 2 :
    if (x != 0) :
        print(“C” , end = ‘ ‘)
print(“D”)
```

Q5. What is the error in following code ? Correct the code:

```
weather = ‘raining’
if weather = ‘sunny’ :
    print(“wear sunblock”)
elif weather = “snow” :
    print(“going skiing”)
else :
    print(weather)
```

Q6. What is the output of the following lines of code ?

```
if str(0) == ‘zero’ :
  print(0);
elif str(0) == ‘0’ :
  print(str(0))
else :
  print(“none of the above”)
```