BCSE101E

Computer Programming: Python

Course Faculty: Rajesh M – SCOPE, VIT Chennai

Module-3 Control Structures in Python

Different patterns in Algorithm

Sequential

- Sequential structure executes the program in the order in which they appear in the program

Selectional (conditional-branching)

- Selection structure control the flow of statement execution based on some condition

Iterational (Loops)

- Iterational structures are used when part of the program is to be executed several times

Sequential Pattern

Example1: Find the average runs scored by a batsman in 4 matches

Algorithm:

Step 1: Start

Step 2: Input 4 scores say runs1,runs2,runs3 and runs4

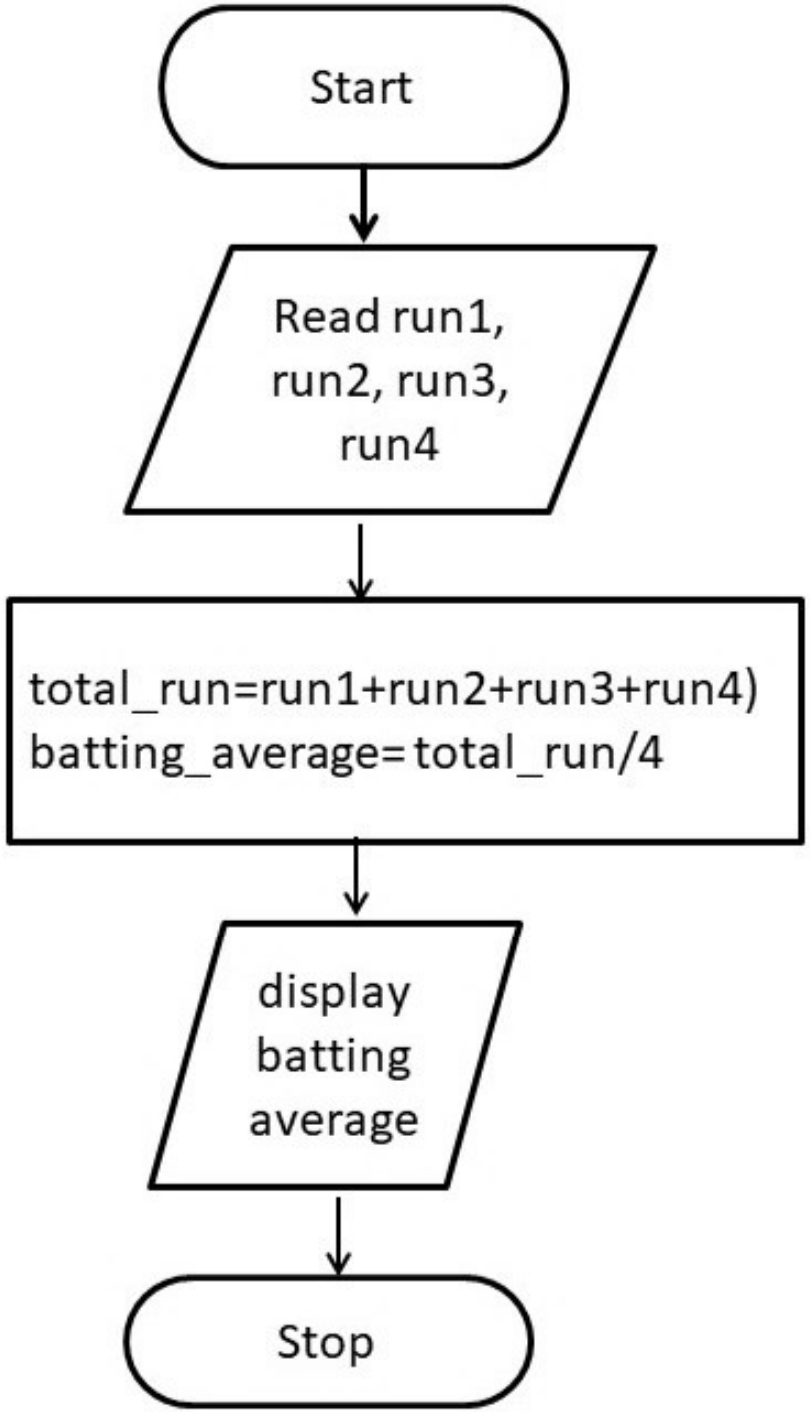
Step 3: Accumulate runs1,runs2,run3,and runs4 and store it in the variable called total_runs

Step 4: Divide total_runs by 4 and find the average

Step 5: Display the average

Step 6: Stop

Flowchart


Start
Read run1, run2, run3, run4
total_run=run1+run2+run3+run4)
batting_average= total_run/4
display batting average
Stop

Pseudo code:

```
Begin
read run1,run2,run3 and run4
compute total_run= run1+run2+run3+run4
compute batting_average= total_run/4
display batting_average
end
```

Batting Average

```
print("Enter four scores")
run1 = int(input())
run2 = int(input())
run3 = int(input())
run4 = int(input())
total_run=(run1+run2+run3+run4)
batting_average= total_run/4
print('batting_average is' ,batting_average)
```

Area of a circle

Step 1 : Start

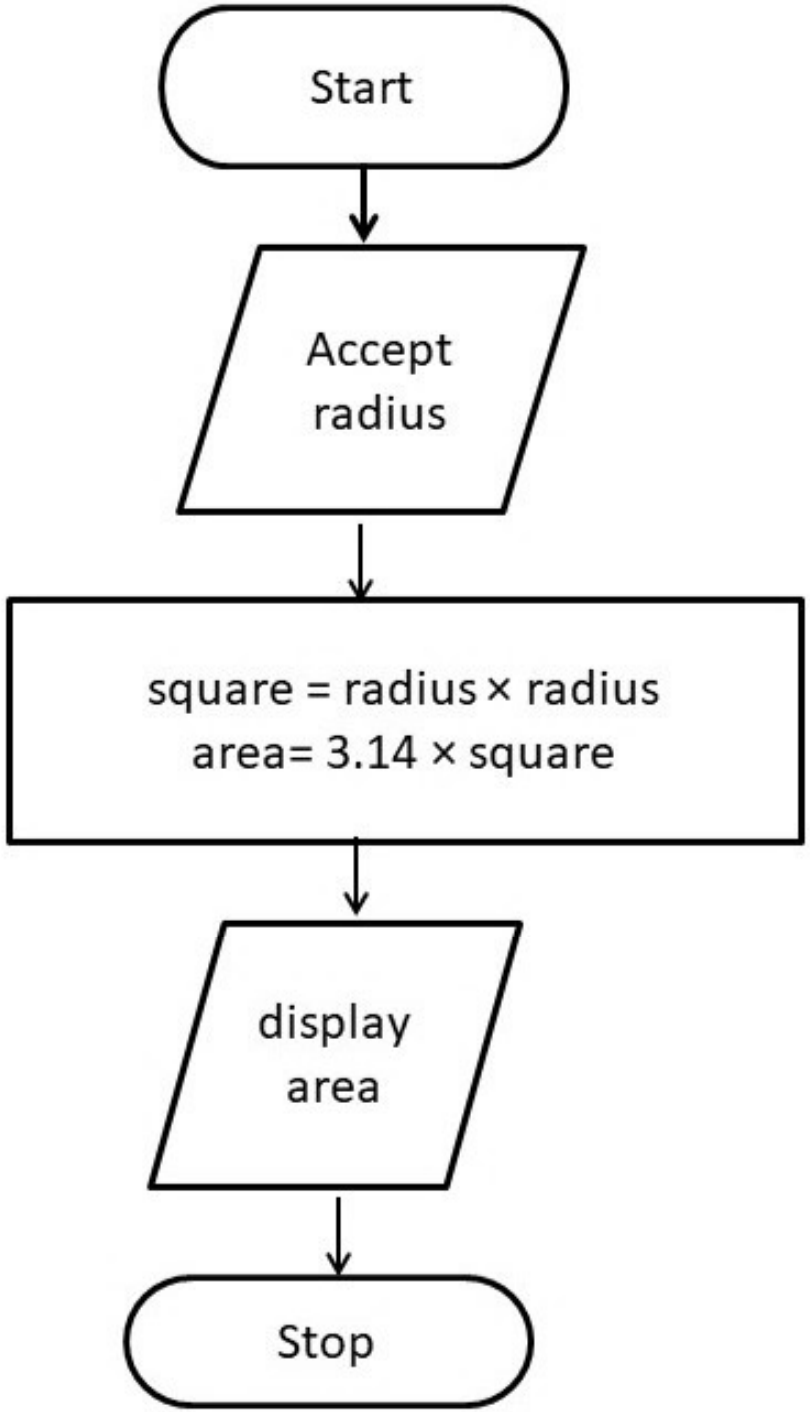
Step 2: Get the input for RADIUS

Step 3 : Find the square of RADIUS and store it in SQUARE

Step 4 : Multiply SQUARE with 3.14 and store the result in AREA

Step 5: Stop

Flowchart


Start
Accept
radius
square = radius × radius
area= 3.14 × square
display
area
Stop

Pseudo code:

```
begin
accept radius
compute square = radius * radius
compute area =  pi * square
display area
end
```

Area of a circle

```
import math
print("Enter radius")
radius=float(input())
area = math.pi*radius*radius
print("area of circle is ", area)
```

Practice Problem - Exercise

An university is setting up a new lab at their premises. Design an algorithm and write Python code to determine the approximate cost to be spent for setting up the lab. Cost for setting the lab is sum of cost of computers, cost of furniture's and labor cost. Use the following formulae for solving the problem:

Cost of computer = cost of one computer * number of computers

Cost of furniture = Number of tables * cost of one table + number of chairs * cost of one chair

Labour cost = number of hours worked * wages per hour

Budget for Lab

Input	Processing	Output
cost of one computer, number of computers, number of tables, cost of one table, number of chairs, cost of one chair, number of hours worked, wages per hour	Budget = Cost of computers + cost of furniture + labour cost Cost of computer = cost of one computer * number of computers Cost of furniture = Number of tables * cost of one table + number of chairs * cost of one chair Labour cost = number of hours worked * wages per hour	Budget for Lab

Python Program

```
print("Enter cost of one computer")
cost_Computer = float(input())
print("Enter num of computers")
num_Computer = int(input())
print("Enter cost of one table")
cost_Table = float(input())
print("Enter num of tables")
num_Tables = int(input())
print("Enter cost of one chair")
cost_Chair = float(input())
print("Enter num of chairs")
num_Chairs = int(input())
print("Enter wage for one hour")
wages_Per_Hr = float(input())
print("Enter num of hours")
num_Hrs = int(input())
```

Python Program

```
cost_Of_Computers = cost_Computer* num_Computer
cost_Of_Furnitures = num_Tables * cost_Table +\
                     cost_Chair*num_Chairs
wages = wages_Per_Hr * num_Hrs
budget = cost_Of_Computers + cost_Of_Furnitures + wages
#format for two decimal places
print ("Budget for Lab ",format(budget,'.2f'))
```

Browsing Problem

Given the number of hours and minutes browsed, write a program to calculate bill for Internet Browsing in a browsing center. The conditions are given below.

(a) 1 Hour Rs.50

(b) 1 minute Re. 1

(c) Rs. 200 for five hours

Boundary condition: User can only browse for a maximum of 7 hours

Check boundary conditions

Browsing Program

Input	Processing	Output
Number of hours and minutes browsed	Check number of hours browsed, if it is greater than 5 then add Rs 200 to amount for five hours and subtract 5 from hours Add Rs for each hour and Re 1 for each minute Basic process involved: Multiplication and addition	Amount to be paid

Pseudocode

```
READ hours and minutes
SET amount = 0
IF hours >=5 then
    CALCULATE amount as amount + 200
    COMPUTE hours as hours – 5
END IF
COMPUTE amount as amount + hours * 50
COMPUTE amount  as amount + minutes * 1
PRINT amount
```

Test Cases

Input

Hours = 6

Minutes = 21

Output

Amount = 271

Processing Involved

Amount = 200 for first five hours

50 for sixth hour

21 for each minute

Test Cases

Input

Hours = 8

Minutes = 21

Output

Invalid input

Processing Involved

Boundary conditions are violated

Already Know

- To read values from user
- Write arithmetic expressions in Python
- Print values in a formatted way

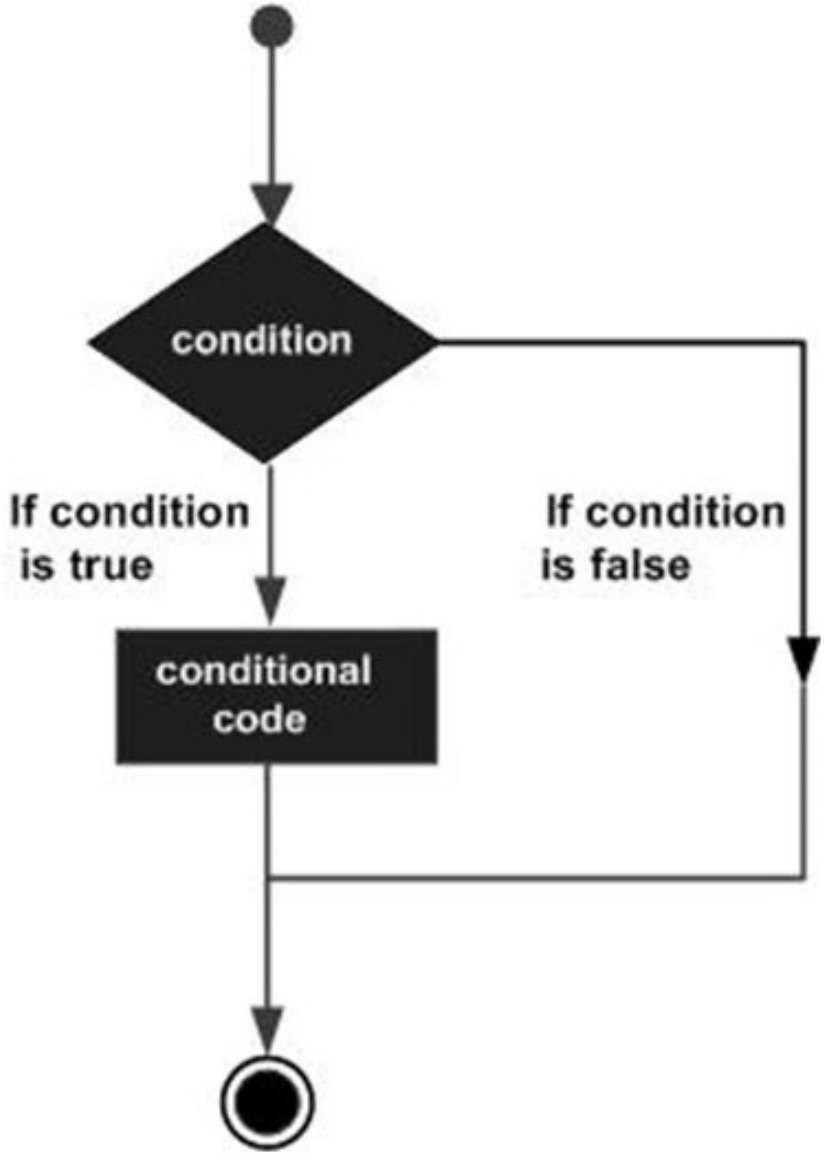
Yet to learn

- Check a condition

Selection pattern

- A **selection control statement** is a control statement providing selective execution of instructions.

Control flow of decision making



If Statement

- An **if statement** is a selection control statement based on the value of a given Boolean expression.
  The if statement in Python is

If statement	Example use
If condition: statements else: statements	If grade >=70: print('pass') else: print('fail')

If Statement

If statement	Example use
If condition: statements else: statements	If grade >=70: print('pass') else: print('fail')

Indentation in Python

- One fairly unique aspect of Python is that the amount of indentation of each program line is significant.
- In Python indentation is used to associate and group statements

Valid indentation		Invalid indentation	
(a) ```if condition: statement statement else: statement statement```	(b) ```if condition: statement statement else: statement statement```	(c) ```if condition: statement statement else: statement statement```	(d) ```if condition: statement statement else: statement statement```

If – Statements in Python

There come situations in real life when we need to do some specific task based on some condition evaluation. In such cases, conditional statements can be used. The following are the conditional statements provided by Python.

- ✓ if
- ✓ if..else
- ✓ Nested if
- ✓ if-elif statements.

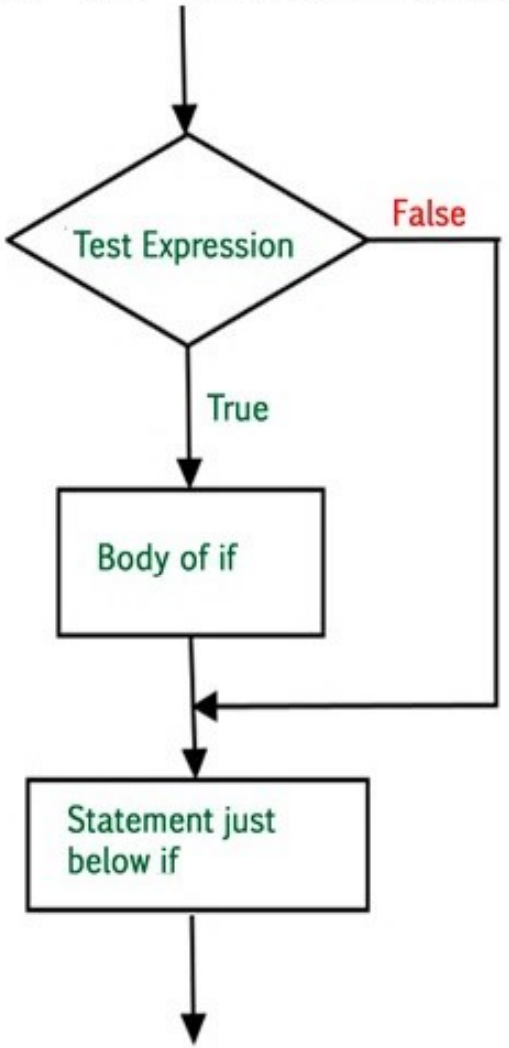
if - Statement

if Statement:

If the simple code of block is to be performed if the condition holds true then, if statement is used. Here if the condition mentioned is true, then the code of block runs otherwise not.

Syntax:

```
if condition:
        # Statements to execute if
        # condition is true (Indentation)
```



Example

```
# if statement example

if 10 > 5:

    print("10 greater than 5")

print("Program ended")
```

Output:

```
10 greater than 5
Program ended
```

Indentation(White space) is used to delimit the block of code. As shown in the above example it is mandatory to use indentation in Python3 coding.

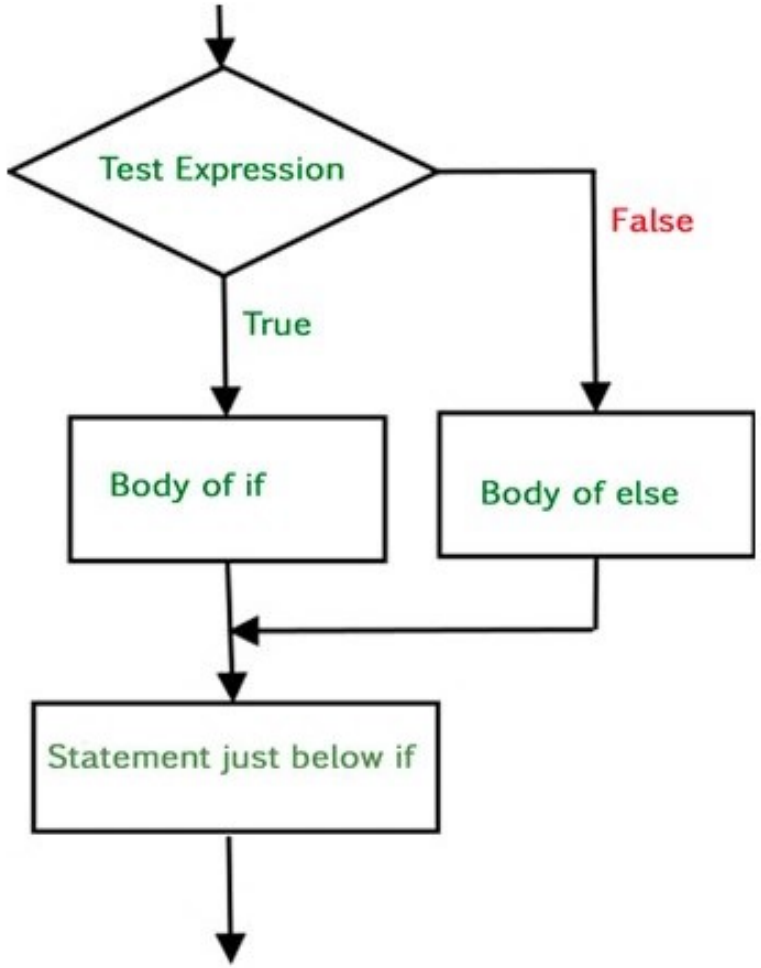
if..else Statement:

if..else Statement:

It has both true and false blocks. Either true or false block would be executed based on condition evaluation of if statement.

Syntax:

```
if (condition):
        # Executes this block if
        # condition is true
else:
        # Executes this block if
        # condition is false
```



Example

```
# if..else statement example

x = 3

if x == 4:

    print("Yes")

else:

    print("No")
```

IF-else construct

Syntax:

```
if(condition):
    statements
else:
    statements
```

Example:

```
N=int(input("Enter n:"))
if(N%2==0):
    print(N,"is even")
else:
    print(N,"is Odd")
```

Example

You can also chain if..else statement with more than one condition.

```
# if..else chain statement
letter = "A"

if letter == "B":
  print("letter is B")

else:

  if letter == "C":
    print("letter is C")

  else:

    if letter == "A":
      print("letter is A")

    else:
      print("letter isn't A, B and C")
```

Output:

```
letter is A
```

Nested if Statements

- There are often times when selection among more than two sets of statements (suites) is needed.
- For such situations, if statements can be nested, resulting in **multi-way selection**.

Nested if statements	Example use
`if condition:` `  statements` `else:` `  if condition:` `  statements` `  else:` `  if condition:` `  statements` `  etc.`	`if grade >= 90:` `  print('Grade of A')` `else:` `  if grade >= 80:` `  print('Grade of B')` `  else:` `  if grade >= 70:` `  print('Grade of C')` `  else:` `  if grade >= 60:` `  print('Grade of D')` `  else:` `  print('Grade of F')`

Else if Ladder

```
if grade >= 90:
    print('Grade of A')
elif grade >= 80:
    print('Grade of B')
elif grade >= 70:
    print('Grade of C')
elif grade >= 60:
    print('Grade of D')
else:
    print('Grade of F')
```

Multiple Conditions

- Multiple conditions can be check in a ‘if’ statement using logical operators ‘and’ and ‘or’.
- Python code to print ‘excellent’ if mark1 and mark2 is greater than or equal to 90, print ‘good’ if mark1 or mark2 is greater than or equal to 90, print ‘need to improve’ if both mark1 and mark2 are lesser than 90

```
if mark1>=90 and mark2 >= 90:
    print('excellent')
if mark1>=90 or mark2 >= 90:
    print('good')
else:
    print('needs to improve')
```

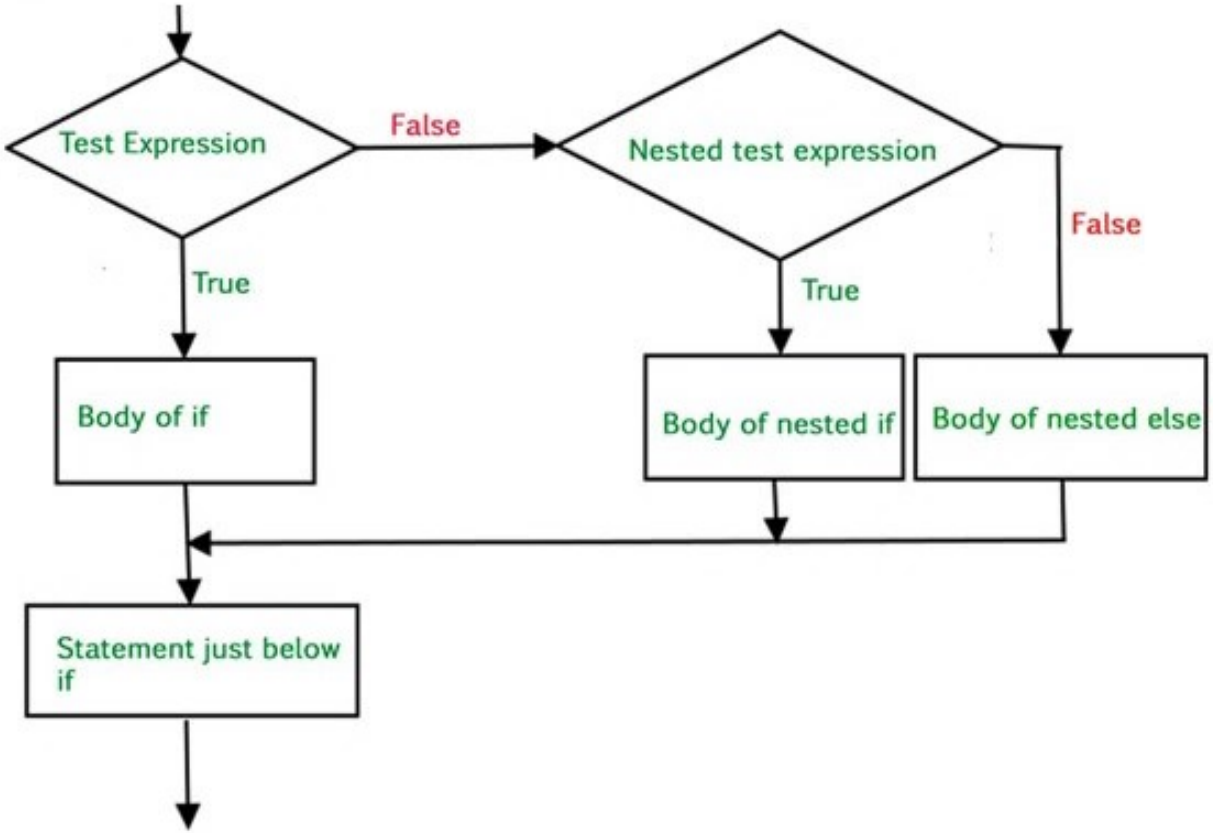
Nested if Statement

Nested if Statement:

if statement can also be checked inside other if statement. This conditional statement is called nested if statement. This means that inner if condition will be checked only if outer if condition is true and by this, we can see multiple conditions to be satisfied.

Syntax:

```
if (condition1):
        # Executes when condition1 is true
        if (condition2):
                # Executes when condition2 is true
        # if Block is end here
# if Block is end here
```



Example

```
# Nested if statement example
num = 10

if num > 5:
    print("Bigger than 5")

    if num <= 15:
        print("Between 5 and 15")
```

Output:

```
Bigger than 5
Between 5 and 15
```

if-elif Statement

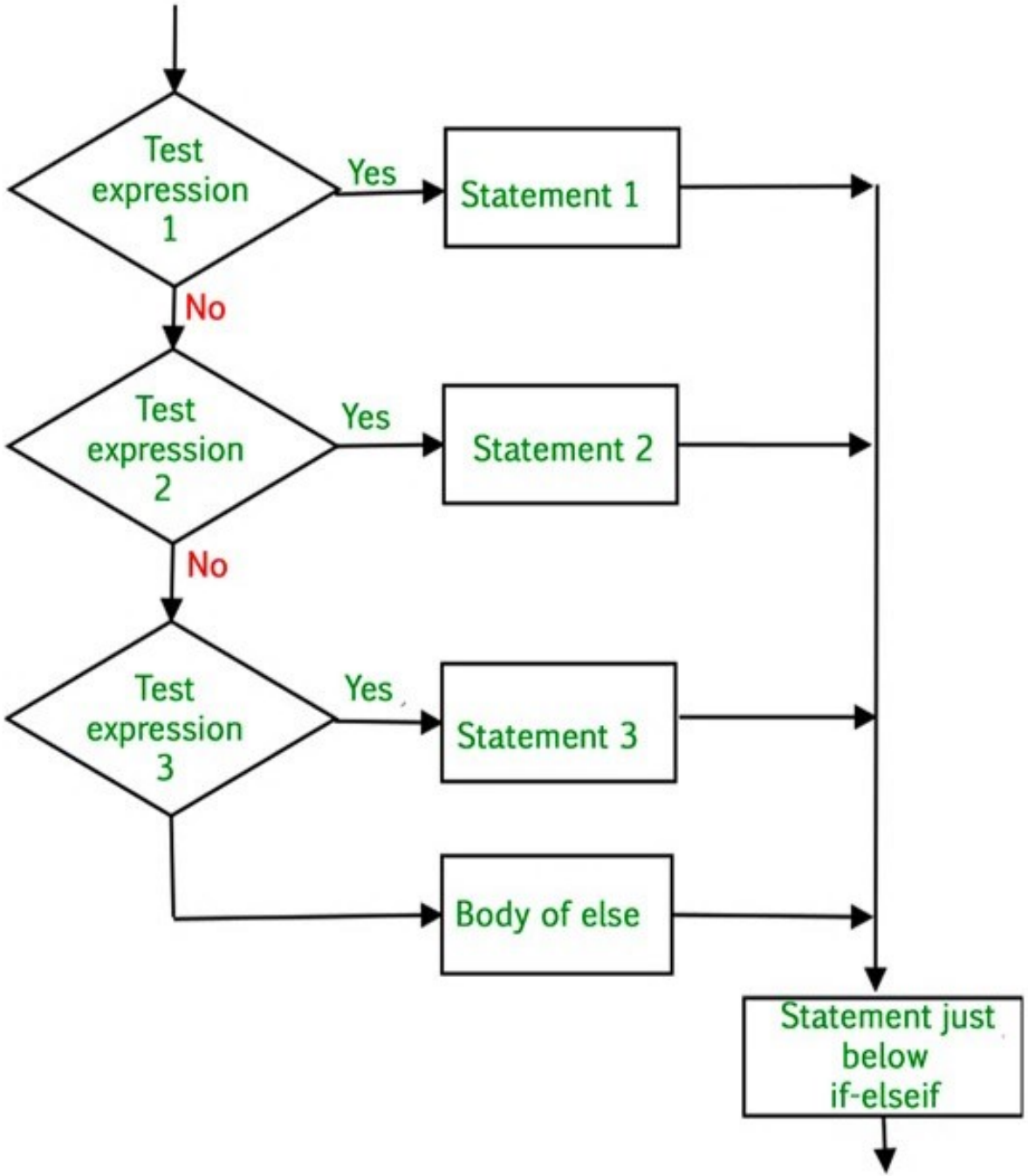
if-elif Statement:

The if-elif statement is shortcut of if..else chain. While using if-elif statement at the end else block is added which is performed if none of the above if-elif statement is true.

Syntax:

```
if (condition):
        statement
elif (condition):
        statement
elif (condition):
        statement
        . .
        . .
        . .
else:
        statement
```

if-elif Statement



Example

```
# if-elif statement exmaple

letter = "A"

if letter == B:
    print("letter is B")

elif letter == "C":
    print("letter is C")

elif num == "A":
    print("letter is A")

else:
    print("letter isn't A, B or C")
```

Browsing Program

```
print("enter num of hours")
hour = int(input())
print("enter num of minutes")
min = int(input())
if(hour>7):
    print("Invalid input")
elif hour>=5:
    amount = 200
    hour = hour - 5
    amount = amount+hour*50+min
    print(amount)
```

Eligibility for Scholarship

Government of India has decided to give scholarship for students who are first graduates in family and have scored average > 98 in math, physics and chemistry. Design an algorithm and write a Python program to check if a student is eligible for scholarship

Boundary Conditions: All marks should be >0

Scholarship Program

Input	Processing	Output
Read first graduate, physcis, chemistry and maths marks	Compute total = phy mark + che mark + math mark Average = total/3 Check if the student is first graduate and average >=98	Print either candidate qualified for Scholarship or candidate not qualified for Scholarship

Algorithm

Step 1 : Start

Step 2: Read first graduate, physcis,chemistry and maths marks

Step 3: If anyone of the mark is less than 0 then print 'invalid input' and terminate execution

Step 3 : Accumulate all the marks and store it in Total

Step 4 : Divide Total by 3 and store it in Average

Step 5 : If student is first graduate Average score is greater than or equal to 98 then print candidate qualified for Scholarship

Else

Print candidate not qualified for scholarship

Stop 6: Stop

Test Cases

Input

First graduate = 1 Phy mark = 98, Che mark = 99,
math mark = 98

Output

candidate qualified for Scholarship

Processing Involved

Total = 295

Average = 98.33

Student is first graduate and average > 98

Test Cases

Input

First graduate = 0 Phy mark = 98, Che mark = 99,
math mark = 98

Output

candidate not qualified for Scholarship

Processing Involved

Total = 295

Average = 98.33

Student is not first graduate but average > 98

Test Cases

Input

First graduate = 1 Phy mark = 98, Che mark = 99,
math mark = 90

Output

candidate not qualified for Scholarship

Processing Involved

Total = 287

Average = 95.67

Student is first graduate but average < 98

```
print('Is first graduate(1 for yes and 0 for no')
first = int(input())
print('Enter Physics Marks')
phy_mark = float(input())
print('Enter Chemistry Marks')
che_mark=float(input())
print('Enter Math Marks')
mat_mark=float(input())
total_mark= phy_mark+che_mark+mat_mark
```

```
if(phy_mark <0 or che_mark <0 or mat_mark<0):
    print('Invalid input')
else:
    average = total_mark/3
    if first==1 and average >= 98 :
        print('candidate qualified for  Scholarship')
    else:
        print('candidate not qualified for  Scholarship')
```

Algorithm for Largest of Three numbers

Step1: Start

Step2: Read value of a, b and c

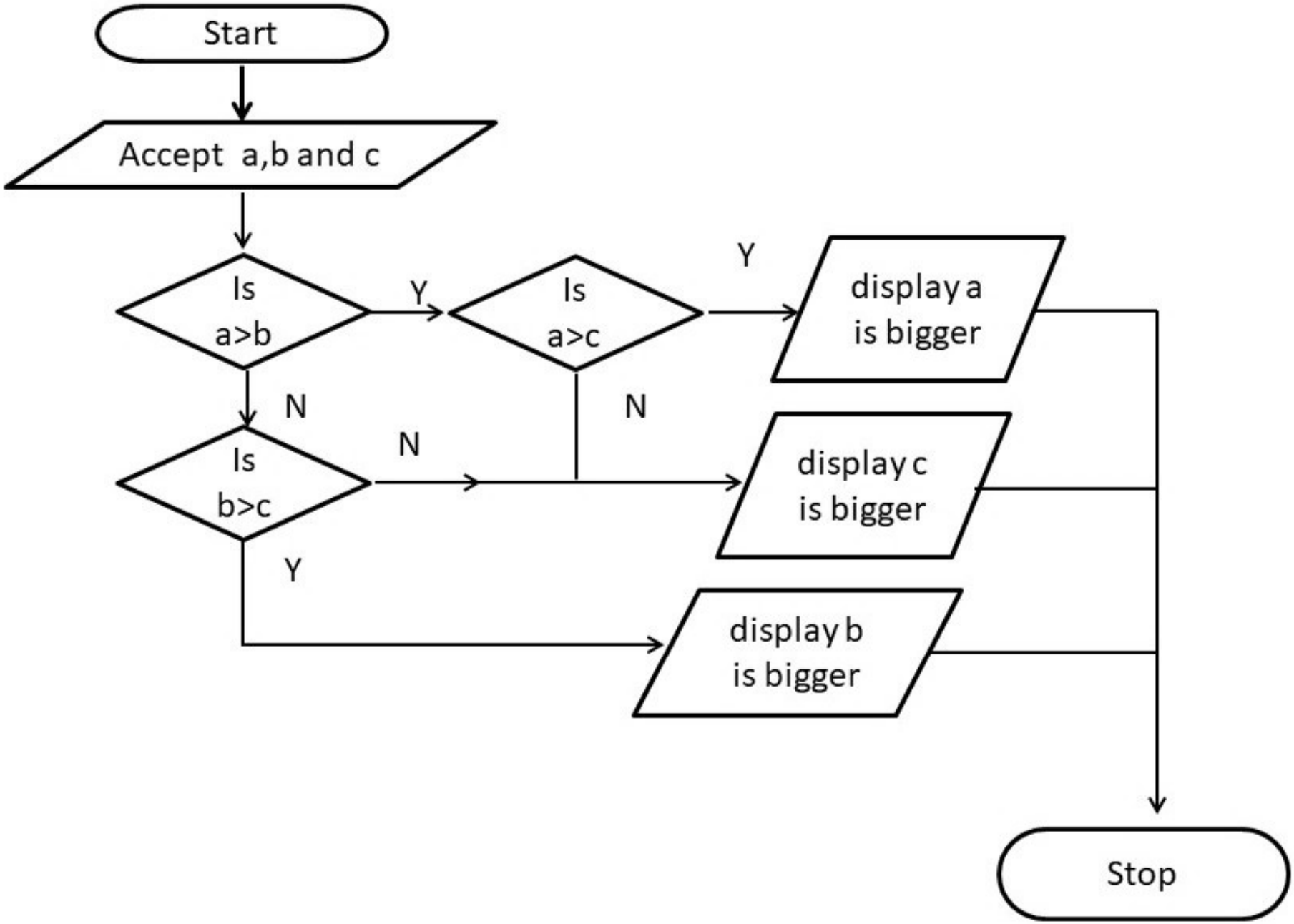
Step3: If a is greater than b then

compare a with c and if a is bigger then say a is biggest else say c is biggest

else Compare b with c , if b is greater than c say b is biggest else c is biggest

Step 5: Stop

Flowchart


Start
Accept a,b and c
Is a>b
Y
Is a>c
Y
display a is bigger
N
N
Is b>c
N
display c is bigger
Y
display b is bigger
Stop

Test Cases

Input

a = 12, b = 13, c = 14

Output

c is greatest

Processing Involved

B is greater than a but c is greater than b

Test Cases

Input

a = 13, b = 12, c = 14

Output

c is greatest

Processing Involved

a is greater than b but c is greater than a

Test Cases

Input

a = 13, b = 2, c = 4

Output

a is greatest

Processing Involved

a is greater than b and a is greater than c

Test Cases

Input

a = 3, b = 12, c = 4

Output

b is greatest

Processing Involved

b is greater than a and b is greater than c

```
a = int(input())
b = int(input())
c = int(input())
if a>b:
    if a>c:
        print ('a is greatest')
    else:
        print ('c is greatest')
else:
    if b>c:
        print ('b is greatest')
    else:
        print ('c is greatest')
```

The if/else Ternary Expression

Consider the following statement, which sets A to either Y or Z, based on the truth value of X:

```
if X:
        A = Y
else:
        A = Z
```

new expression format that allows us to say the same thing in one expression:

- A = Y if X else Z

```
>>> A = 't' if 'spam' else 'f'
>>> A
't'
>>> A = 't' if '' else 'f'
>>> A
'f'
```