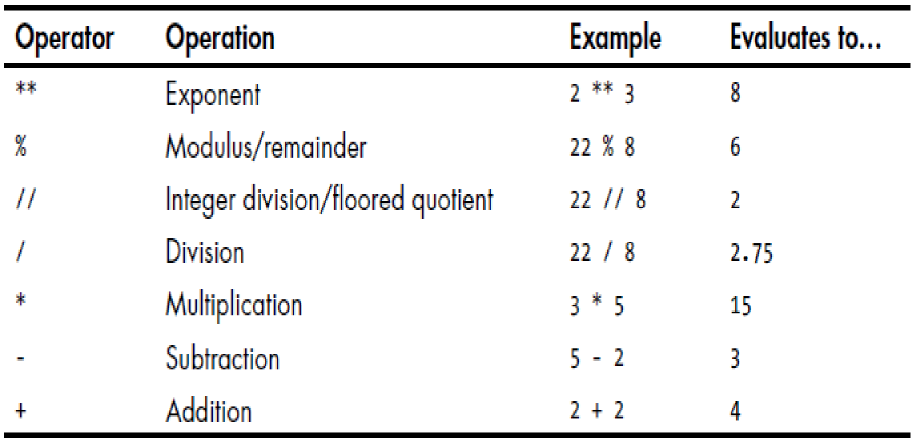
Module wise Question Bank Solution

MODULE – 1 Python Basics

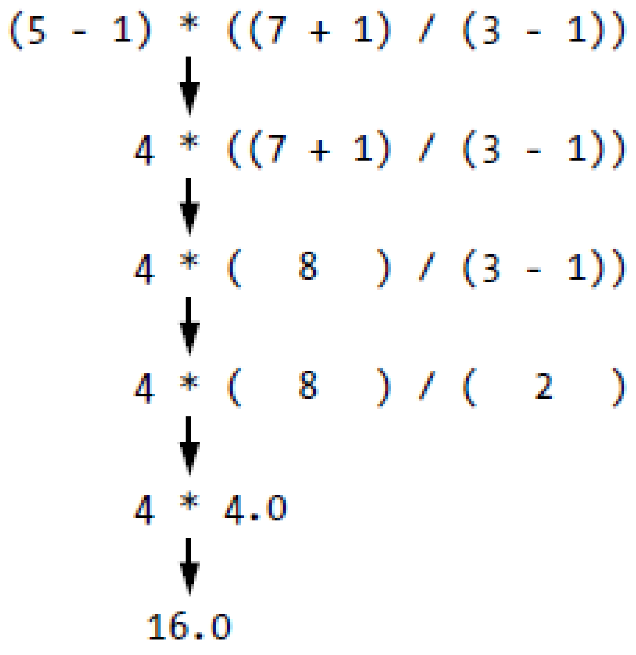
1 **Explain the math operators in Python from highest to lowest Precedence with an example for each. Write the steps how Python is evaluating the expression (5 - 1) * ((7 + 1) / (3 - 1)) and reduces it to a single value.**

- ✓ The operator is a symbol which tells the compiler to do specific mathematical or logical function.
- ✓ In python, a single value with no operators is also considered an expression, though it evaluates only to itself.
- ✓ The order of operations (also called precedence) of Python math operators is similar to that of mathematics i.e., **, *, /, //, %, +, -
- ✓ The associativity is also similar and followed from left to right.



Operator	Operation	Example	Evaluates to...
**	Exponent	2 ** 3	8
%	Modulus/remainder	22 % 8	6
//	Integer division/floored quotient	22 // 8	2
/	Division	22 / 8	2.75
*	Multiplication	3 * 5	15
-	Subtraction	5 - 2	3
+	Addition	2 + 2	4

- ✓ The steps for evaluating the expression in python is as follows:



```
(5 - 1) * ((7 + 1) / (3 - 1))
        ↓
      4 * ((7 + 1) / (3 - 1))
        ↓
      4 * (   8   ) / (3 - 1))
        ↓
      4 * (   8   ) / (   2   )
        ↓
      4 * 4.0
        ↓
      16.0
```

2 **What are Comparison and Boolean operators? List all the Comparison and Boolean operators in Python and explain the use of these operators with suitable examples.**

✓ **Comparison operators:** These are used to compare two values and evaluate down to a single Boolean value.

Operator	Meaning
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

✓ **Boolean Operators:** The three Boolean operators (and, or, and not) are used to compare Boolean values.

and operator: The and operator evaluates an expression to True if both Boolean values are True; otherwise, it evaluates to False.

Table 2-2: The and Operator's Truth Table

Expression	Evaluates to...
True and True	True
True and False	False
False and True	False
False and False	False

```
>>> True and True
True
>>> True and False
False
```

ii. *or operator:* The or operator valuates an expression to True if either of the two Boolean values is True. If both are False, it evaluates to False.

Table 2-3: The or Operator's Truth Table

Expression	Evaluates to...
True or True	True
True or False	True
False or True	True
False or False	False

```
>>> False or True
True
>>> False or False
False
```

iii. *not operator:* The not operator operates on only one Boolean value (or expression). The not operator simply evaluates to the opposite Boolean value. Much like using double negatives in speech and writing, you can nest not operators ❶, though there‟s never not no reason to do this in real programs.

Table 2-4: The not Operator's Truth Table

Expression	Evaluates to...
not True	False
not False	True

```
  >>> not True
  False
❶ >>> not not not not True
  True
```

3 **Write a python program to find the area of rectangle and area of a circle. Print the results. Take input from user.**

area of rectangle

```
length = float(input('Enter length of rectangle'))
breadth = float(input('Enter breadth of rectangle'))
area = length*breadth
print("The area of rectangle is ",area)
```

Output:

```
Enter length of rectangle: 4.0
Enter breadth of rectangle:2.0
Area of rectangle is : 8.0
```

area of a circle

```
radius = float(input('Enter radius of circle'))
area = 2*3.14*radius
print("The area of circle is ",area)
```

Output:

```
Enter radius  of a circle: 5
Area of circle is : 78.5
```

4	**What are Control Statements? Explain the syntax of all flow control statements along with flow chart and examples.** There are four types of flow control statements: *1.* if Statement *2.* else Statement *3.* elif Statement	

1. *if Statements:*

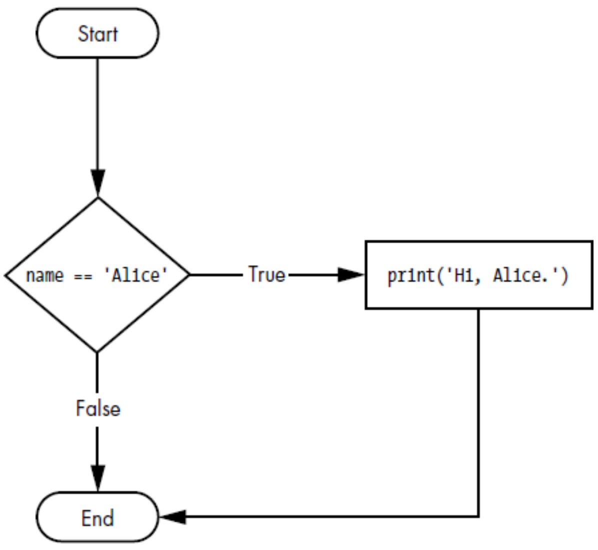
- An if statement‟s clause (that is, the block following the if statement) will execute if the statement‟s condition is True. The clause is skipped if the condition is False.
- In plain English, an if statement could be read as, "If this condition is true, execute the code in the clause."
- In Python, an if statement consists of the following:

1. The if keyword
2. A condition (that is, an expression that evaluates to True or False)
3. A colon
4. Starting on the next line, an indented block of code (called the if clause)

- Example:

```
if name == 'Alice':
    print('Hi, Alice.')
```

- Flowchart:



2. *else Statements:*

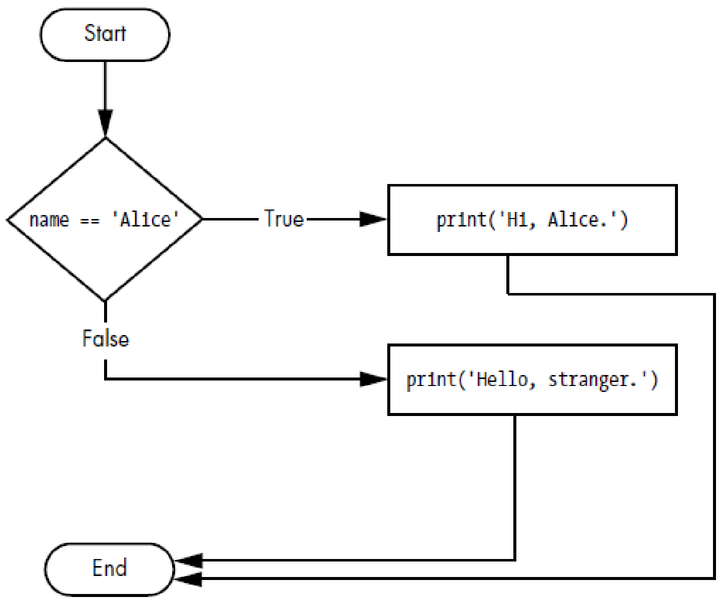
- An if clause can optionally be followed by an else statement. The else clause is executed only when the if statement‟s condition is False.
- In plain English, an else statement could be read as, "If this condition is true, execute this code. Or else, execute that code."
- An else statement doesn‟t have a condition, and in code, an else statement always consists of the following:

1. The else keyword
2. A colon
3. Starting on the next line, an indented block of code (called the else clause)

- Example:

```
if name == 'Alice':
    print('Hi, Alice.')
else:
    print('Hello, stranger.')
```

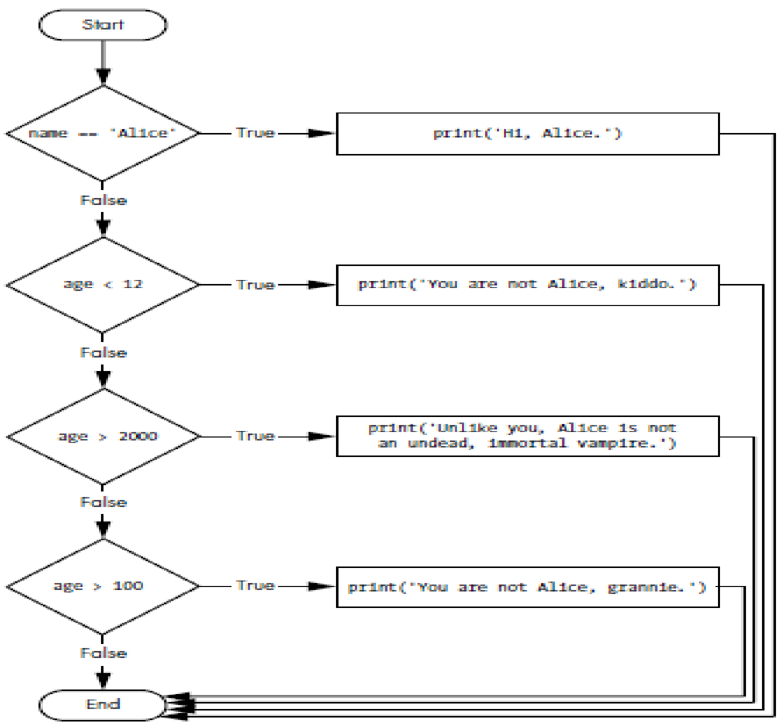
- Flowchart:



3. *elif Statements:*

- If we have more than one condition then elif statement is used.
- The elif statement is an “else if” statement that always follows an if or another elif statement.
- It provides another condition that is checked only if all of the previous conditions were False.
- In code, an elif statement always consists of the following:
  1. The elif keyword
  2. A condition (that is, an expression that evaluates to True or False)
  3. A colon
  4. Starting on the next line, an indented block of code (called the elif clause)

- Flowchart:


Start
name == 'Alice'
True
print('Hi, Alice.')
False
age < 12
True
print('You are not Alice, kiddo.')
False
age > 2000
True
print('Unlike you, Alice is not an undead, immortal vampire.')
False
age > 100
True
print('You are not Alice, grannie.')
False
End

- Example:

```
if name == 'Alice':
    print('Hi, Alice.')
elif age < 12:
    print('You are not Alice, kiddo.')
elif age > 2000:
    print('Unlike you, Alice is not an undead, immortal vampire.')
elif age > 100:
    print('You are not Alice, grannie.')
```

5 Illustrate with an example python program string concatenation and string replication

String concatenation

- The meaning of an operator may change based on the data types of the values next to it
- For example, + is the addition operator when it operates on two integers or floating-point values.
- However, when + is used on two string values, it joins the strings as the string concatenation operator.

```
>>> 2 + 2
4
```

```
>>> 'Alice' + 'Bob'
'AliceBob'
```

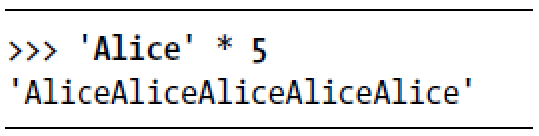
- If we try to use the + operator on a string and an integer value, Python will not know how to handle this, and it will display an error message

```
>>> 'Alice' + 42
Traceback (most recent call last):
  File "<pyshell#26>", line 1, in <module>
    'Alice' + 42
TypeError: Can't convert 'int' object to str implicitly
```

String replication

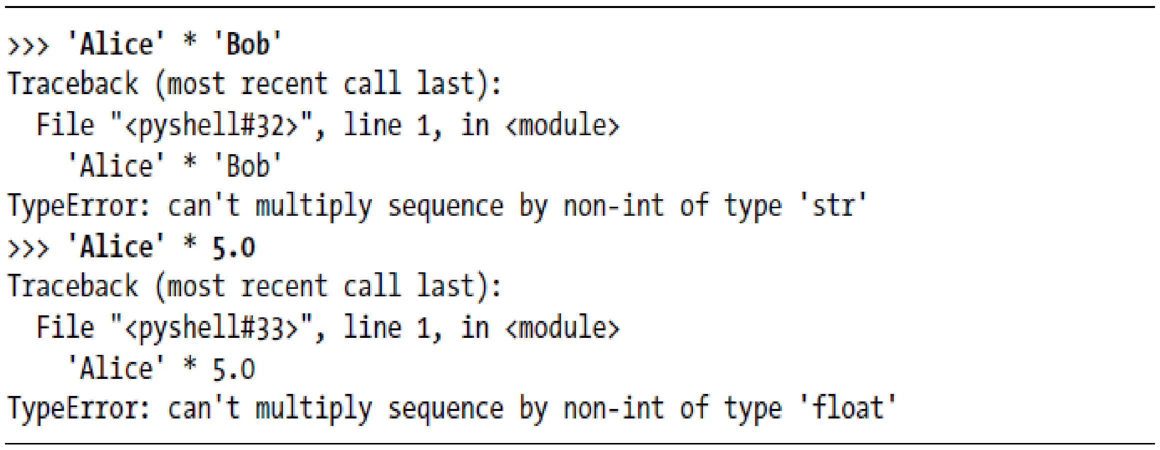
The * operator is used for multiplication when it operates on two integer or floating-point values.

- But, when the * operator is used on one string value and one integer value; it becomes the string replication operator



```
>>> 'Alice' * 5
'AliceAliceAliceAliceAlice'
```

The * operator can be used with only two numeric values (for multiplication) or one string value and one integer value (for string replication). Otherwise, Python will just display an error message.



```
>>> 'Alice' * 'Bob'
Traceback (most recent call last):
  File "<pyshell#32>", line 1, in <module>
    'Alice' * 'Bob'
TypeError: can't multiply sequence by non-int of type 'str'
>>> 'Alice' * 5.0
Traceback (most recent call last):
  File "<pyshell#33>", line 1, in <module>
    'Alice' * 5.0
TypeError: can't multiply sequence by non-int of type 'float'
```

6. What is nested-if statement? Illustrate with an example python program

An if-Statement or an if-else statement present within another if-statements or if –else statements is called nested-if statements. When an action has to be performed based on many decisions involving various types of expressions and variables, then the nested if-statement is used it is also called multi-way decision statement

Example 1: if a>b:

```
if a>c:
  print("A is greater\n")
  sys.exit()
```

Example 2:
```
if a>b:
        if a>c:
           print(“A is greater \n”)
           sys.exit()
      else:
         print(“C is greater \n”)
         sys.exit()
```

7. Write the design and python program to check a whether a given number is odd number or even number with an output

Design:
```
   print(“Even”) if n%2==0
   print(“Odd”) otherwise
```

Program:
```
 n= int(input(“Enter an integer : ”)
    if(n%2)== 0:
       print(“Even”)
       sys.exit()
   else:
   print(“Odd”)
```

Output:

Test case 1:
```
Enter a number: 6
Even
```

Test case 2:
```
Enter a number:3
Odd
```

8. What are identifiers? Describe the rules of identifiers and List out the valid and invalid identifiers

Definition: The names given to various program elements such as variables,functions,lists etc are called identifiers

Rules to be followed

1. It can be only one word.
2. It can use only letters, numbers, and the underscore (_) character.
3. It can‘t begin with a number.
4. Reserved words(also called keywords) can’t be used as identifiers

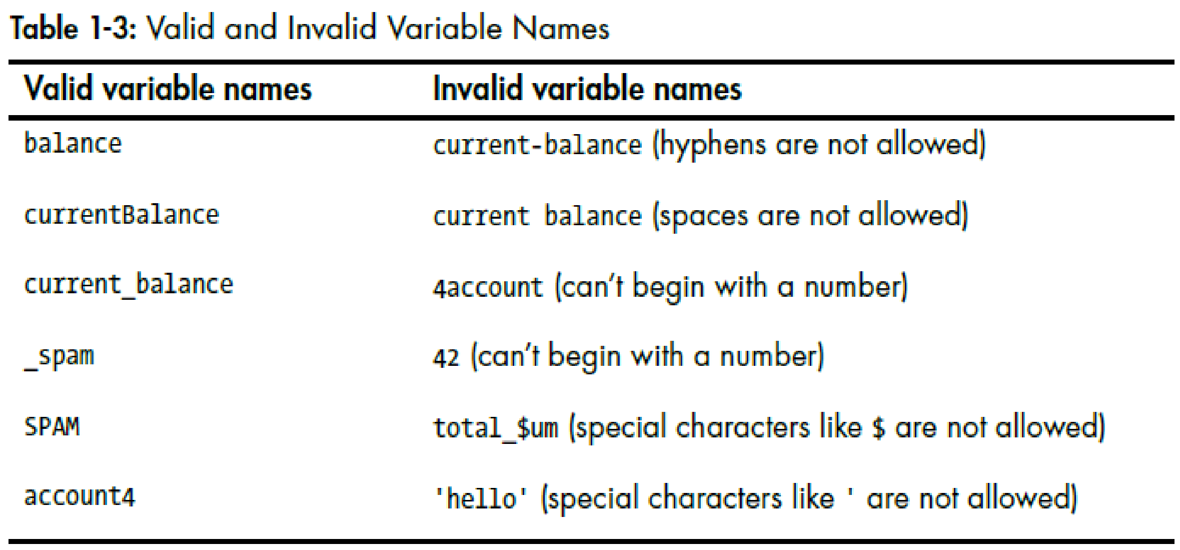


Table 1-3: Valid and Invalid Variable Names

Valid variable names	Invalid variable names
balance	current-balance (hyphens are not allowed)
currentBalance	current balance (spaces are not allowed)
current_balance	4account (can't begin with a number)
_spam	42 (can't begin with a number)
SPAM	total_$um (special characters like $ are not allowed)
account4	'hello' (special characters like ' are not allowed)

9. write python program along with a design to check whether the given triangle is isosceles, equilateral or scalene

Design:

Print “Equilateral”

if a==b and b==c

exit

print “isosceles”

if a==b or b==c or c==a

exit

print “scalene”

if a!=b and b!=c and c!=a

exit

Program:

```
import sys
a= int(input("Enter the first  side of a triangle: ")
b= int(input("Enter the Second side of a triangle: ")
c= int(input("Enter the third  side of a triangle: ")
if a==b and b==c:
          print("equilateral")
          sys.exit()
if a==b or b==c or c==a
        print("iscoscles")
        sys.exit()
if a!=b and b!=c and c!=a
      print("scalene")
```

Output:

Test case 1:

```
Enter the first  side of a triangle:10
Enter the Second side of a triangle:10
Enter the Third side of a triangle:10
Equilateral
```

Test case 2:

```
Enter the first  side of a triangle:10
Enter the Second side of a triangle:8
Enter the Third side of a triangle:10
Isosceles
```

Test case 3:

```
Enter the first  side of a triangle:10
Enter the Second side of a triangle:8
Enter the Third side of a triangle:9
Scalene
```

10. Write a python program to find largest of three numbers using else-if statements with test cases.

<u>Syntax:</u>

B-Statements

```
if condition-1:
        statement-1
elif condition-2:
       statement-2
elif condition-3:
       statement-3
…………..
…………..
elif condition-- n-1:
       statement--n-1
else:
      statement - n
```

A-statements

Program:

```
a=int(input("Enter the first number:")
b=int(input("Enter the second number:")
c=int(input("Enter the third number:")
if a>b and a>c:
      big=a
      print("big")
elif b>a and b>c:
      big=b
      print("big")
else:
    big=c
    print("big")
print("maximum(%d, %d, %d) =%d" % (a, b, c ,big)
```

Test cases:

```
Run 1: Enter the first number:10
       Enter the second number:8
       Enter the third number:5
       Maximum(10,8,5)=10

Run 2: Enter the first number:4
       Enter the second number:9
       Enter the third number:2
       Maximum(4,9,2)=9

Run 3: Enter the first number:1
       Enter the second number:7
       Enter the third number:3
       Maximum(1,7,3)=7
```

11. What are loop statements? With syntax write a python program to find a factorial of a number using while loop

Definition:
A set of statements that are repeatedly executed till the condition is met or till number of times are called loop statements.

syntax:

```
B-Statements
Initialization
while expression:
      statement T1;
      statement T2;
      ………………
      ……………….
 AStatements
```

Program:

```
n= int(input("Enter the value of n:"))
i=1
factorial=1
while i<n:
    factorial=factorial*i
    i=i+1
print(factorial)
```

Output:

```
Enter the value of n: 5
120
```

12. Illustrate an algorithm and write a python program to print GCD of two numbers

Algorithm GCD(m,n):

M	N
10	6
4	6
2	2
2	2

Algorithm GCD(m,n):

//Input:m,n positive integer

```
if( m>n)
    m=m-n
else:
    n=n-m         while(m!=n)
```

Program:

```
m = int(input("Enter the value for m:"))
n =  int(input(Enter the value for n:")

a=m ,b=n;

while m!=n:
      if m>n:
           m=m-n
      else:
          n=n-m

print("GCD (%d,%d) = %d" %(a,b,m))
```

Output:

```
GCD(10,6)= 2
```