Loop Statements (for,while)

Statements	Syntax	Example	Meaning

while	while (Condition):
	statement(s)
		count = 0
		while (count < 3):
		count = count+1
		print("Hello Bennettians")
		Output:
		Hello Bennettians
		Hello Bennettians
		Hello Bennettians
			while loop is used for iterators

for	for iterator_var in sequence:
	statements(s)
		l = ["bennett", "for", "bennettians"]
		for i in l:
		print(i)
		Output:
		bennett
		for
		bennettians
			for can be used to iterate over iterators and a range.

nested-for	for iterator_var in sequence:
	for iterator_var in sequence:
	statements(s)
	statements(s)
		for i in range(1, 5):
		for j in range(i):
		print(i, end=' ')
		print( )
		Output:
		1
		2 2
		3 3 3
		4 4 4 4
			Python programming language allows to use for loop inside another for loop.

nested-while	while expression:
	while expression:
	statement(s)
	statement(s)
		i = 1
		j = 5
		while i < 4:
		while j < 8:
		print(i, ",", j)
		j = j + 1
		i = i + 1
		Output:
		1 , 5
		2 , 6
		3 , 7
			Python programming language allows to use while loop inside another while loop.

1. Predict the output:

```
count = 0
while True:
   print("Bennett")
   print(count +=1)
```

```
Ans: Bennett (infinite loop)
```

2. Predict the output:

```
num = 10
while num > 6:
   print(num)
   num = num-1
   print(num)
print("Loop End")
```

```
Ans:    10
        9
        9
        8
        8
        7
        7
        6
        Loop End
```

3. Predict the output:

```
sum = 0
for val in range(1, 6):
        sum = sum + val
print(sum)
```

```
Ans: 15
```

4. Print the following pattern

```
*
* *
* * *
* * * *
* * * * *
```

```
Ans: for i in range(0, 5):
         for j in range(0, i+1):
                print("* ",end="")
         print("\r")
```

5. Write a program using while loop to generate the first 10 natural numbers and their sum.

```
Ans:

i=0
sum = 0
while i<10:
   print(i,end=" ")
   i+=1
   sum +=i
print("sum: ",sum)
```

6. Program to find prime numbers in a given range using for loop: (*range 25 to 50*)

```
Ans:
for num in range(25, 50):
   # all prime numbers are greater than 1
   if num > 1:
       for i in range(2, num):
           if (num % i) == 0:
               break
       else:
           print(num)
```

7. Write a Program to Print the Fibonacci sequence, using while loop

Note: In mathematical terms, the sequence F_n of Fibonacci numbers is defined by the

$F_n = F_{n-1} + F_{n-2}$

with

$F_0 = 0$ *and* $F_1 = 1.$

Input: How many terms? (e.g., 7)
Output: Fibonacci sequence: 0 1 1 2 3 5 8

Ans:

```
nterms = int(input("How many terms? "))
# first two terms
n1, n2 = 0, 1
count = 0
# check if the number of terms is valid
if nterms <= 0:
   print("Please enter a positive integer")
# if there is only one term, return n1
elif nterms == 1:
   print("Fibonacci sequence upto",nterms,":")
   print(n1)
# generate fibonacci sequence
else:
   print("Fibonacci sequence:")
   while count < nterms:
       print(n1, end=" ")
       nth = n1 + n2
       # update values
       n1 = n2
       n2 = nth
       count += 1
```

8. Write a program to take the input from user (*e.g.*, *num* = *5*), and compute the factorial, using for loop.

```
Ans:
num = 5
# To take input from the user
#num = int(input("Enter a number: "))

factorial = 1

# check if the number is negative, positive or zero
if num < 0:
   print("factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of",num,"is",factorial)
```

9. Write a Python program to find the sum of the following series for the given values of *x* and *n*. (*input x = 2, n = 5*)

sum = *1 + x + x^2 / 2 + x^3/3 + ... + x^n/n*

Ans:

```
x = 2
n = 5
sum = 1
for i in range(1, n + 1):
        sum = sum + ((x**i)/i)

print(round(sum, 2))
```

Control Statements (Continue,Break,Pass)

Statements	Example	Meaning

Continue	for char in ‘Pythn’:
	if (char == ‘y’):
	continue
	print(“Current character: “, char)
	Output:
	Current character: P
	Current character: t
	Current character: h
	Current character: n
		When the program encounters continue statement, it will skip the statements which are present after the continue statement inside the loop and proceed with the next iterations.

break	for char in ‘Python’:
	if (char == ‘h’):
	break
	print(“Current character: “, char)
	Output:
	Current character: P
	Current character: y
	Current character: t
		The break statement is used to terminate the loop containing it, the control of the program will come out of that loop.

pass	for char in ‘Python’:
	if (char == ‘h’):
	pass
	print(“Current character: “, char)
	Output:
	Current character: P
	Current character: y
	Current character: t
	Current character: h
	Current character: o
	Current character: n
		Pass statement is python is a null operation, which is used when the statement is required syntactically.

1. Predict the output:

```
for num in [22, 11, 19, 66, 14, 99, 55]:
        if num%2 == 0:
             continue
        print(num)
```

```
Ans:
11
19
99
55
```

2. Given a list iterate it and display numbers which are divisible by 5 and if you find number greater than 150 stop the loop iteration

list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]

Output:
15
55
75
150

```
Ans:
list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]
list2 = []
for i in list1:
    if (i > 150):
        break
    if i %5 ==0:
        list2.append(i)

print(list2)
```

3. Predict the output

```
s = "bennett"
for i in s:
    if i == 'n':
        print('Pass executed')
        pass
    print(i)
```

Ans:

```
b
e
Pass executed
n
Pass executed
n
e
t
t
```

4. Predict the output

```
for i in range(4):
    for j in range(4):
        if j==2:
            break
        print("The number is ",i,j);
```

Ans:

```
The number is  0 0
The number is  0 1
The number is  1 0
The number is  1 1
The number is  2 0
The number is  2 1
The number is  3 0
The number is  3 1
```

5. Predict the output

```
my_list = ['C/C++', 'JAVA', 'Python', 'Lisp', 'Ruby', 'Python']
i = 0

while True:
    print(my_list[i])
    if (my_list[i] == 'Python'):
        print('Found the name Python')
        break
        print('After break statement')
    i += 1

print('After while-loop exit')
```

Ans:

```
C/C++
JAVA
Python
Found the name Python
After while-loop exit
```

6. Fill in the black (i.e., ??, here), and also predict the output.

```
for n in range(2, 10):
  for x in range(2, n):
         if n % x == 0:
                  print(n, 'equals', x, '*', n//x)
                  break
         else:
              print(n, 'is a ?? number')
```

```
Ans:
        ?? -> prime
        3 is a prime number
        4 equals 2 * 2
        5 is a prime number
        5 is a prime number
        5 is a prime number
        6 equals 2 * 3
        7 is a prime number
        7 is a prime number
        7 is a prime number
        7 is a prime number
        7 is a prime number
        8 equals 2 * 4
        9 is a prime number
        9 equals 3 * 3
```

7. Program to calculate the sum of 4 numbers, and calculate sum until user enters positive number

Ans:

```
sum = 0
while True:
    print("Enter the number:");
    num = eval(input())
     #If number is -ve then control
     # came out from the while loop
    if(num < 0):
         break

    sum += num # sum = sum + number
    i +=1

    if( i>10 ): #maximum 10 numbers
        break

print("Sum ", sum)
```