code: CSA0810

subject: python programming

Name: Jeevan

Reg No: 192211714

ASSIGNMENT-III

Iteration in python such as state, while, for, break, continue and pass with Examples.

In python you can use various iteration and control flow constructs like 'while', 'for', 'break', 'continue' & 'pass' to control the flow of program

Iteration or looping statement:

An Iterative statement allows us to execute a statement multiple times

→ Repeated execution of a set of statement is called Iteration on looping.

Types of Iterative statements:

→ while loop

→ for loop

→ Nested loop.

While:

A while loop executes a block of statement again and again until the condition gets false.

→ The while Keyword is followed by test expression

→ following the header is on indented body.

Syntax:

The 'while' loop continues to execute a block of code as long as a given condition is "True".

Examples of 'while' loop:

```
n = int (input(Enter a number: ))
i = 1
```

```
fact=1
while (i<n):
    fact = fact*i
    i=i+1
    print("factorial is", fact)
```

using the while loop will describe that if i is less than or equal to n.

→print statement will give the output.

```
Input = 5
```

output:

```
the factorial is =120
```

```
count=0
while count < 3
    print(f"count is {count}")
    count = count + 1
```

the output:

```
0, 1, 2.
```

Using for loop:

for loop is used to iterate a sequence of elements (list, tuple, string) for a specified number of times.

→ for loop in python starts with the keyword 'for' following by an orbitary variable name, which holds its value in the following sequence object.

Syntax:

```
for iterating variable in sequence
            statement.
```

Example for 'for loop':

```
fruits = ["apple", "banana", "cherry"]
for fruit in fruits:
    print(f"current fruit: {fruits}")
```

→A sequence represent a list or a string.

→The iterating variable takes the first item in sequence.

output:

```
current fruit : apple
current fruit : banana
current fruit : cherry.
```

Nested loop:

python program allows using on loop inside another loop.

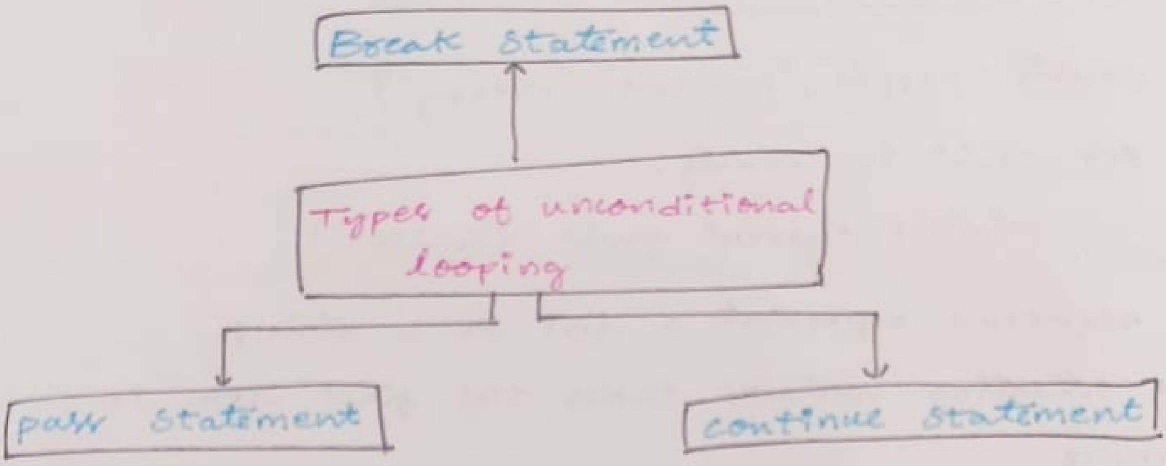
→ for example, using a while loop or a for loop inside of another while for loop.

Syntax:

```
for iterating variable sequence:
    for iterating variable sequence:
        innerloop statement
    outer loop statement.
```

Unconditional Statement:

A situation in which need to exit a loop completely when an external condition is triggered. (or) need to skip a part of the loop.



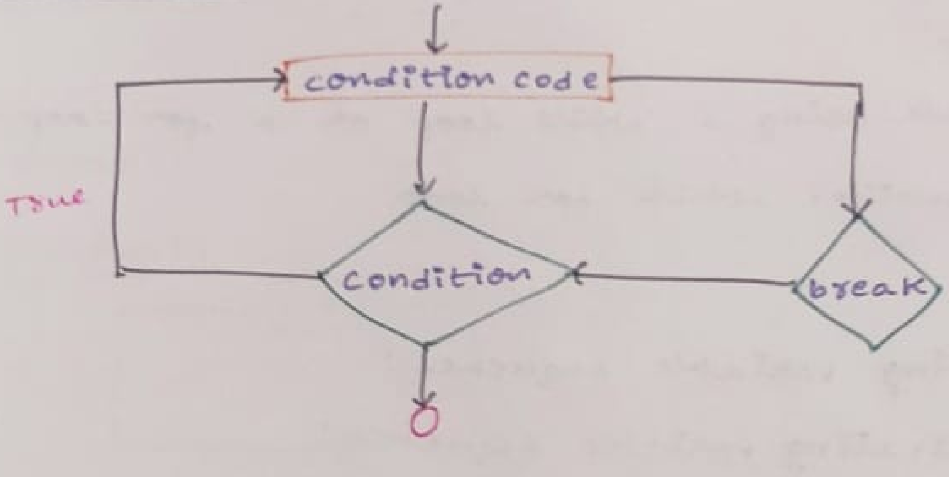
continue statement:

A continue statement returns the control of the beginning of the loop statement.

Syntax:

```
continue
```

flowchart:



The continue statement is used to skip the current iteration of a loop & move to next one.

Example:

```
for num in range(5):
    If num==2:
        continue
    print(num)
```

continue statement will move forward If num is equal to 2.

output:

```
0, 1, 3, 4
```

Break statement:

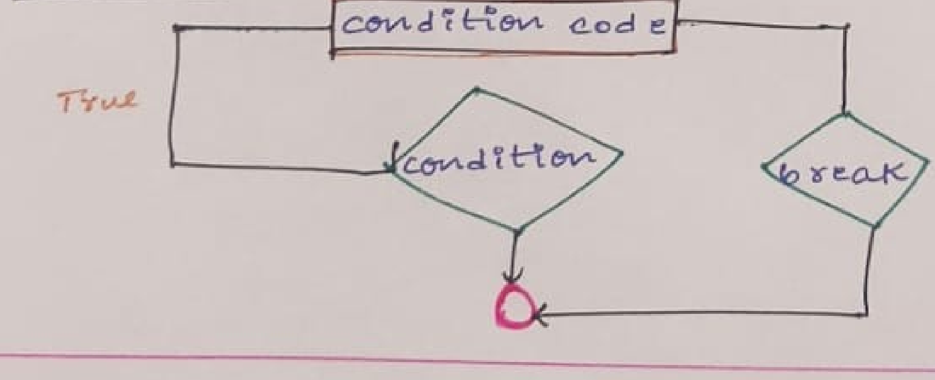
A break statement terminates the current loop & transfers the execution of statement immediately following the loop.

The break statement is used when some external condition is triggered.

Syntax:

```
Break
```

flowchart:



The break statement is used to exist a loop permatarily when a certain condition is met.

Example:

```
for num in range(10):
    if num == 5:
        break
        print(num)
```

output:

```
0, 1, 2, 3, 4
```

Pass Statement:

The pass statement is a Null operation & notthing happens when it executed.

It can be used when a statement is required syntactically but one program require no action.

Syntax:

```
pass
```

Example:

```
for letter in "python"
    if letter == 'n':
        pass
        print(letter)
    print("bye")
```

output:

```
python
'Bye'.
```