AC Patil
College of Engineering

Jawahar Education Society's
A. C. Patil College of Engineering, Kharghar
Navi Mumbai 410210

Student Name: CHETAN INGALE

PRN No.: 221111030

Course Name: C.S.E. (IoT CS BC)

Course code: CSL405

Year: S.E.

Semester: IV

Roll No.: 17

Experiment Evaluation Sheet

Experiment No.: 2

Experiment Name:
To implement Control Statments in Python

Sr No.	Evaluation Criteria	Marks (Out of 9)	Performance Date	Correction Date and Signature of Instructor
1	Experiment Performance			
2	Journal Performance			
3	Punctuality			
Total				

Aim : To implement control statments like for, while, if, elif etc. in Python.

Software required : Python

Theory :

Loops are employed in Python to iterate over a section of code continually. Control statements are designed to serve the purpose of modifying a loop's execution from its default behaviour. Based on a condition, control statements are applied to alter how the loop executes. In this tutorial, we are covering every type of control statement that exists in Python.

if Statements:
The if statement is arguably the most used statement to control loops. There can be zero or more elif parts, and the else part is optional. The keyword 'elif' is short for 'else if', and is useful to avoid excessive indentation. An if … elif … elif … sequence is a substitute for the switch or case statements found in other languages.

for Statements:
The for statement in Python differs a bit from what you may be used to in C or Pascal. Rather than always iterating over an arithmetic progression of numbers (like in Pascal), or giving the user the ability to define both the iteration step and halting condition (as C), Python's for statement iterates over the items of any sequence (a list or a string), in the order that they appear in the sequence.

while Statement:
he while statement is used for repeated execution as long as an expression is true. This repeatedly tests the expression and, if it is true, executes the first suite; if the expression is false (which may be the first time it is tested) the suite of the else clause, if present, is executed and the loop terminates. A break statement executed in the first suite terminates the loop without executing the else clause's suite. A continue statement executed in the first suite skips the rest of the suite and goes back to testing the expression.

Code 2.a :

```
a = int(input("Enter first number :- "))
b = int(input("Enter Second number :- "))
c = int(input("Enter Third number :- "))
smallest= 0
if(a<b)and(a<c):
    smallest = a
if(b<a)and(b<c):
    smallest = b
if(c<a)and(c<b):
    smallest = c
print("Smallest Number is ",smallest)
```

Output 2.a :

```
student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/02Practical$ python3 smallestNumber.py
Enter first number :- 25
Enter Second number :- 86
Enter Third number :- 32
Smallest Number is  25
```

Code 2.b :

```
a = int(input("Enter first number :- "))
b = int(input("Enter Second number :- "))
c = int(input("Enter Third number :- "))
largest = 0
if(a>b)and(a>c):
    largest = a
elif(b>c):
    largest = b
else:
    largest = c
print("Largest Number is ",largest)
```

Output 2.b :

```
student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/02Practical$ python3 largestNumber.py
Enter first number :- 25
Enter Second number :- 86
Enter Third number :- 32
Largest Number is  86
```

Code 2.C :

```
n = int(input("Enter n: "))
sum = 0
i = 1
while i <= n:
    sum = sum + i
    i = i+1
print("The sum is", sum)
```

Output 2.C :

```
student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/02Practical$ python3 addnNatural.py
Enter n: 569
The sum is 162165
```

Code 2.D :

```
lowerValue = int(input ("Enter the Lowest Range Value: "))
upperValue = int(input ("Enter the Upper Range Value: "))
print ("The Prime Numbers in the range are: ")
for number in range (lowerValue, upperValue + 1):
   if number > 1:
      for i in range (2, number):
         if (number % i) == 0:
            break
      else:
         print (number)
```

Output 2.D :

```
student@csiot-ThinkCentre-M70s:~/CHETAN_I_007/PythonLab/02Practical$ python3 primeNumbers.py
Enter the Lowest Range Value: 0
Enter the Upper Range Value: 10
The Prime Numbers in the range are:
2
3
5
7
```

Conclusion :

From this practical we learn how to implement control statments like for, while, if, elif etc. in Python.