NAME:SUPRIYA SUBHASH VIRKAR CLASS:SE-3 ROLL NO:58

Experiment No. 2
To implement Conditional Statements and Loop in python
Date of Performance:
Date of Submission:

Experiment No. 2

Title: To implement Conditional Statements and Loop in python

Aim: To study, and implement Conditional Statements and Loop in python

Objective: To introduce Conditional Statements and Loop in python

Theory:

1. Conditional Statements

There comes situations in real life when we need to do some specific task and based on some specific conditions and, we decide what should we do next. Similarly there comes a situation in programming where a specific task is to be performed if a specific condition is True. 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.

Let us go through all of them.

if Statement

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

if..else Statment

In conditional if Statement the additional block of code is merged as else statement which is performed when if condition is false.

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.

if-elif Statment

The if-elif statement is shoutcut 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.

2. Looping in python

Python programming language provides following types of loops to handle looping requirements. Python provides three ways for executing the loops. While all the ways provide similar basic functionality, they differ in their syntax and condition checking time.

While Loop:

In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied. And when the condition becomes false, the line immediately after the loop in program is executed.

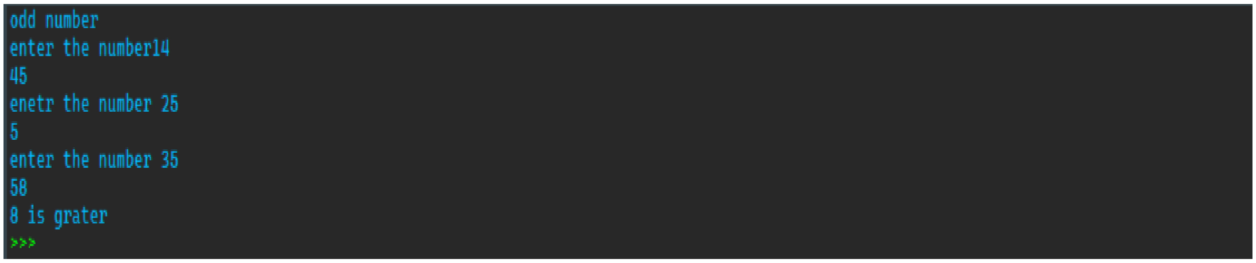
for in Loop:

For loops are used for sequential traversal. For example: traversing a list or string or array etc. In Python, there is no C style for loop, i.e., for (i=0; i<n; i++). There is "for in" loop which is similar to for each loop in other languages. Let us learn how to use for in loop for sequential traversals.

Code:

```
num=float(input("enter a number"));
if(num%2==0):


    print("even number");
else:
    print("odd number");






    number1=input(input("enter the number1"));
    number2=input(input("enetr the number 2"));
    number3=input(input("enter the number 3"));



    if(number1>number2 and number1>number3):
        print(number1 ,"is grater");

    elif(number2>number1 and number2>number3):
        print(number2,"is grater");
    else:
          print(number3,"is grater");

for i in range (0,11)

 print(i);
```

Output:



```
odd number
enter the number14
45
enetr the number 25
5
enter the number 35
58
8 is grater
>>>
```

Conclusion:

- Conditional statements and loops are fundamental constructs in programming that allow you to control the flow of execution and perform repetitive tasks efficiently.

- They help in writing more flexible and dynamic programs by enabling decision-making and iteration.

- Proper indentation is crucial in Python to define blocks of code within conditional statements and loops.

- Understanding when to use each type of loop or conditional statement is essential for writing clean, readable, and efficient code.

- Practice and familiarity with these constructs are key to mastering their usage and leveraging their power in solving various programming problems.