**Student Name:**

**Roll No:**

**Section:**

Experiment No. 05

*Lab 05 –User Defined Functions.*

**Lab Objectives:**

1. Simple while loop

2. User Defined Functions (UDF)

**1. Simple *while* Loop**

A while loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

**Syntax**

The syntax of a while loop in Python programming language is −

while expression:

statement(s)

else:

statement(s)

Here, statement(s) may be a single statement or a block of statements with uniform indent. The condition may be any expression, and true is any non-zero value. The loop iterates while the condition is true.

When the condition becomes false, program control passes to the line immediately following the loop.

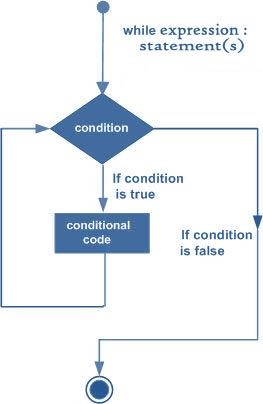
In Python, all the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements. As shown in figure 1, a simple structure of while loop for understanding of logic behind the while loop.

**Student Name:**

**Roll No:**

**Section:**

Figure 1: A simple while loop architecture



A key point of the while loop is that the loop might not ever run. When the condition is tested and the result is false, the loop body will be skipped and the first statement after the while loop will be executed. **Program 1:** Practicing with simple while loop, giving them the final condition of loop.

**Code:**

count = 0

f = eval(input("Enter your final condition where you want to break the loop: "))

while (count < f):

print("The value of count:", count)

count = count+1

print("I am using while loop", count, "time.")

**Output:**

**Student Name:**

**Roll No:**

**Section:**

**Program 2:** Practicing with simple while loop with infinite iterations as the condition is always true.

**Code:**

var = 1

while var == 1 : # This will generate an infinite loop, as condition is always true num = eval(input("Enter a number :"))

print("You entered: ", num)

print ("End of Loop!")

# use Ctrl+C to break the loop

**Output:**

**Program 3:** Write a program which takes the limit for while loop condition and sum the total amount.

**Code:**

n = eval(input("Enter the value to execute the while loop:"))

sum = 0 i = 1

while i <= n:

sum = sum + i i = i+1

print("The sum is", sum)

**Output:**

**Student Name:**

**Roll No:**

**Section:**

**2. *While* with *else* Condition:**

An else statement can be combined with an if statement. An else statement contains the block of code that executes if the conditional expression in the if statement resolves to 0 or a FALSE value.

The else statement is an optional statement and there could be at most only one else statement following if.

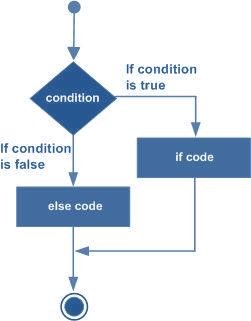


Figure 2: The flow structure of *if* with *else* statement

**Program 4:** Practicing with simple while loop with else condition.

**Code:**

i = 0

f = 10 while i < f:

print (i, " is less than final condition")

i = i + 1 else:

print (i, " is not less than final condition")

**Output:**

**Student Name:**

**Roll No:**

**Section:**

**3. Difference between *for* and *while* loop**

For loop is used to iterate over elements of a sequence. It is often used when you have a piece of code which you want to repeat "n" number of time.

While Loop is used to repeat a block of code. Instead of running the code block once, It executes the code block multiple times until a certain condition is met.

**4. Functions in Python**

Functions are an essential part of the Python programming language: you might have already encountered and used some of the many fantastic functions that are built-in in the Python language or that come with its library ecosystem. However, as a Computer Scientist, you’ll constantly need to write your own functions to solve problems that your projects or problems poses to you.

We will introduce you to functions in Python. We will learn the following parts under function:

i. Functions vs Methods

ii. Parameters vs Arguments

iii. How to define a function: User-Defined Functions (UDFs)

You use functions in programming to bundle a set of instructions that you want to use repeatedly or that, because of their complexity, are better self-contained in a sub-program and called when needed. That means that a function is a piece of code written to carry out a specified task. To carry out that specific task, the function might or might not need multiple inputs. When the task is carried out, the function can or cannot return one or more values.

There are three types of functions in Python:

**1. Built-in-Function:**

Built-in functions, such as help() to ask for help, min() to get the minimum value, print() to print an object to the terminal.

**2. User-Defined Functions:**

User-Defined Functions (UDFs), which are functions that users create to help them out.

**3. Anonymous Functions:**

Anonymous functions, which are also called lambda functions because they are not declared with the standard def keyword.

**Student Name:**

**4.1 Functions vs Methods**

**Roll No:**

**Section:**

A method refers to a function which is part of a class. You access it with an instance or object of the class. A function doesn’t have this restriction: it just refers to a standalone function. This means that all methods are functions but not all functions are methods.

**4.2 Parameters vs Arguments**

Parameters are the names used when defining a function or a method, and into which arguments will be mapped. In other words, arguments are the things which are supplied to any function or method call, while the function or method code refers to the arguments by their parameter names. As shown in the following program where we will make a function f which takes one argument name x.

**Program 5:** Write a function *f* which takes one argument x, it will square the value of x and add 1 in it then return the answer to user.

**Code:**

def f(x):

res = x\*\*2 + 1 return res

**Output:**

**4.3 How to define a function: User-Defined Functions (UDFs)**

User defined functions (UDFs) are normally defined by the keyword def followed by the following format:

*def <function name> (argument1, argument2,<multiple arguments>):*

*body of the function return <value>*

Keep in mind the indentation is very important in programming python code.

**Student Name:**

**Roll No:**

**Section:**

**Output:**

**Student Name:**

**Roll No:**

**Section:**

return outText

**Output:**cryptText.append(crypting)

newLetter = lowercase[crypting]

outText.append(newLetter)

**Student Name:**

**Programming Exercise**

**Roll No:**

**Section:**

Q.1 Write down a Python program, using While loop that generates Odd no’s in between 1 to 100.

Q.2 Write down a Python Program using While loop to generate the following outputs

1. \*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. \*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

1. \*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

Q.3 Write down a python program having one function for calculating factorial of a no.

And call that function within a While loop to generate factorial of numbers from 0 to 10