# 

# Department of Computing

# CS114: Fundamentals of Programming

# Class: BESE 9 AB

# Lab 08: Nested Loops & Functions

**CLO4: Adopt the latest IDEs and other supplementary tools to aid code implementation and management**

# Instructor: Ms. Hania Aslam

# Date: November 9th , 2018

# Time: 9:00am -12:00pm and 02:00pm -05:00pm

**Lab 08: Nested Loops & Functions**

**Introduction**

The purpose of this lab is to get familiar with usage of nested loops and functions in Python.

**Tools/Software Requirement**

Python IDLE

**Description:**

**Loops in Python:**

Loops are an essential part of any programming language. Likewise, for and while loops are an important part of Python as well. In the previous labs you have learned to use While and For loops in Python. We can nest these loops together to design solutions of complex problems.

**Nested for Loops in Python**

The simple definition of a nested loop is a loop inside a loop (i.e. placing a for loop inside another for loop). We can use any number of nested loops, the depth of nested loop depends on the complexity of a problem.

We can have different combinations of nested loops:

1. While loop within a while loop.
2. For loop within a while loop.
3. For loop within a for loop.
4. While loop within a for loop

Consider the following example:

**Nested Loop Example:**

**for** row in range (0, 3 ,1):

**for** col in range (0, 2 ,1):

**prin**t(“row: ",row," col: ",col)

**Functions in Python:**

A functions is a block of organized & reusable code that performs a specific task. Using functions we can write modular programs that have minimal code duplication.

**Description:**

* Function blocks begin with the keyword **def** followed by the function name and parentheses ( ( ) ).
* Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.
* The code block within every function starts with a colon (:) and is indented.
* The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

|  |
| --- |
| >>> def quadratic(a, b, c, x):  first = a \* x \*\* 2  second = b \* x  third = c  return first + second + third  >>> quadratic(2, 3, 4, 0.5) 6.0  >>> quadratic(2, 3, 4, 1.5) 13.0 |

**Lab Tasks:**

**Perform the following tasks:**

**Task 1:**

Use simple nested loops to generate all possible coordinates from (0,0) up to (5,5). Name the program BasicNestedLoops.py . Your output must appear in rows and columns as follows.



|  |
| --- |
| Task 1 |
| #Add your Python Script code here. [1.5 Marks]  CODE:    #Add the snap of tasks execution here. [0.5 Mark]  OUTPUT: |

**Task 2:**

Write a Python program which prints the following pattern of numbers using nested loops. You are not required to make the boxes!!

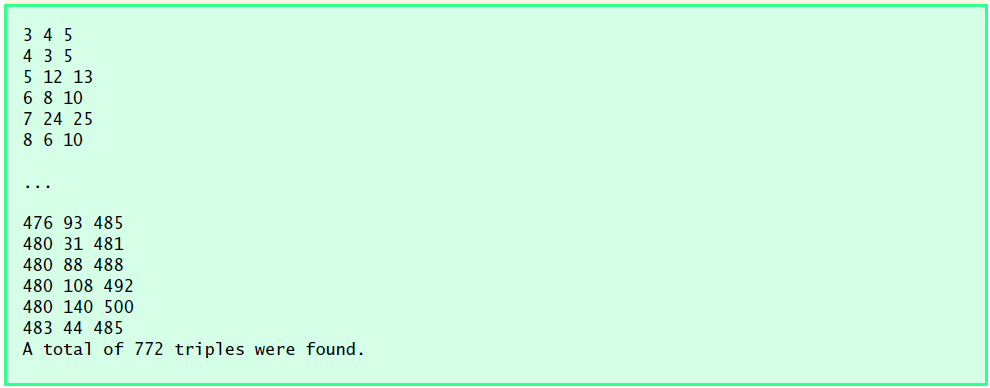
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | 1 |
|  |  | 2 | 1 |
|  | 3 | 2 | 1 |
| 4 | 3 | 2 | 1 |

**Hints:** The outer loop will be used for rows. Two inner loops will be used inside the outer loop. The first inner loop will adjust the spacing, while the other inner loop will print the numbers accordingly. Remember, we can also specify a negative step size in range function to generate a reverse sequence!

|  |
| --- |
| Task 2 |
| #Add your Python Script code here. [1.5 Marks]  CODE:    #Add the snap of tasks execution here. [0.5 Mark]  OUTPUT: |

**Task 3:** (Pythagorean Triples) A right triangle can have sides that are all integers. The set of three integer values for the sides of a right triangle is called a Pythagorean triple. These three sides must satisfy the relationship that the sum of the squares of two of the sides is equal to the square of the hypotenuse. Find all Pythagorean triples for side1, side2, and the hypotenuse all no larger than 500. Use a triple-nested while loop that simply tries all possibilities. This is an example of “brute force” computing. It is not aesthetically pleasing to many people. But there are many reasons why these techniques are important as you will learn in more advanced computer science courses, there are large numbers of interesting problems for which there is no known algorithmic approach other than sheer brute force.

|  |
| --- |
| Task 3 |
| #Add your Python Script code here. [1.5 Marks]    #Add the snap of tasks execution here. [0.5 Mark]  OUTPUT:    And so on… |



**Task 4:**

Write a function for a grading app which would take a number score and convert it to a letter grade, with the following requirements.

* Grade of 90 and above is an A
* Grade of 80 to 89 is a B
* Grade of 70 to 79 is a C
* Grade of 60 to 69 is a D
* Grade of 59 or below is an F

Test if your function is working fine by calling it.

|  |
| --- |
| Task 4 |
| #Add your Python Script code here. [1.5 Marks]  CODE:    #Add the snap of tasks execution here. [0.5 Mark]  OUTPUT: |

**Task 5:**

Write a program that consists of the following conversion functions:

1. Miles to Km converter

2. Celsius to Fahrenheit converter

3. ft. to meters converter

4. Yards to meters converter

Your program must keep on prompting the user to select the conversion function of their choice until user enters a sentinel value to exit the program. Once user input is received appropriate result should be displayed to the user!

|  |
| --- |
| Task 5 |
| #Add your Python Script code here. [1.5 Marks]  CODE:    #Add the snap of tasks execution here. [0.5 Mark]  OUTPUT: |

**Deliverables**

Compile a single Word document by filling in the solution/answer part (as directed) along with the snapshots. Name your submission file as given below and submit this Word file on LMS before the deadline.

**Name – Registration No. – Section**

**Name: Hamid Ayub**

**Regt. No: 12933118**

**Section: BESE\_9B**

**Grade Criteria**

This lab is graded. Min marks: 0. Max marks: 10.

|  |  |  |
| --- | --- | --- |
| **Activity** | **Minimum** | **Maximum** |
| Documentation with clearly defined understanding of the lab task and approach | Fail | Pass |
| Task 1 | 0 | 2 |
| Task 2 | 0 | 2 |
| Task 3 | 0 | 2 |
| Task 4 | 0 | 2 |
| Task 5 | 0 | 2 |