Assignment SU9

```
Reminder: Add your name, surname, and student number as a
comment line at the top of each of your programs. Use comments
to name and explain each script.
```

9.1 **Submit your Python script file (*.py) on eFundi Assignments;** named:

`StudentNumber_SU9-1 (include your student number).`

Aim

The aim of this assignment is for you to apply higher order thinking skills to showcase your ability to reach the outcomes of this study unit. This assignment is project based; This mean that you will receive some minimum requirements for developing your own Python script. You will need to come up with a concept and write the code.

Instructions

The table provides the minimum requirements, as an example you may use 3 (three) variables in your program, but you must have at least 2 (two) variables. Take special care to include comments that explain what the script needs to do. This is an opportunity to showcase your skill and creativity.

Assignment SU9-1 REQUIREMENTS

CONCEPT	**MINIMUM REQUIREMENTS**
INPUT	Two (2) variables
PROCESSING	One (1) if statement (multiway)
	One (1) else statement
	One (1) loop, either for or while
OUTPUT	Neatly displayed

9.2 **Submit your Python script file (*.py) on eFundi Assignments;** named:

`StudentNumber_SU9-2 (include your student number).`

Problem

You need to provide a multiplication table (from 1 to 12) for the value entered by the user, utilising a for loop.

Instructions

1. Ask the user to enter an integer value. The value entered by the user must greater than 1 (one), therefore, apply input validation by using a while loop. Ask the user to enter a value until a valid entry is made.

2. Display the multiplication table based on the user-input. Example of output if the user enters the value of 5:

```
1 x 5 = 5
2 x 5 = 10
3 x 5 = 15
4 x 5 = 20
5 x 5 = 25
6 x 5 = 30
7 x 5 = 35
8 x 5 = 40
9 x 5 = 45
10 x 5 = 50
11 x 5 = 55
12 x 5 = 60
```

3. Lastly, modify the script to determine the sum of the totals from the multiplication table, print the total.

Make sure you supply headings to ensure that your input and output make sense.

9.3 **Submit your Python script file (*.py) on eFundi Assignments;** named:

`StudentNumber_SU9-3 (include your student number).`

Problem

Write a Python script that prompts the user for an email address, extract the name and domain from the address using a loop, and print the results.

Instructions

1. Ask the user to enter an email address.
2. Loop through the characters in the email address.
   Hint: Remember that Python strings can be treated as arrays of characters, so you can use array indexing and slicing to extract substrings from the email address.
   E.g., `emailString[index]` with index meaning the position within the string, represented by an integer value.
3. Print out the name and domain as output.

Notes:

- The `len()` function in Python returns the length of a string (number of characters contained in the string). E.g., `len(name)` will return the value three (3) if the name contained "Sam".
- Using block brackets with two integer values separated by a colon, e.g., [*start*:*end*], allows one to select parts of a string in Python, named a substring.
  E.g., `emailString[:5]` => will return the first characters from a string, for a total of five (5) characters.
  Or, `emailString[6:]` => will return the sixth character until the last character.

Assessment rubric

Mark allocation rubric			
Item		**Mark allocation**	
9.1	INPUT (two (2) variable)		/1
	PROCESSING (one (1) if statement – multiway)		/1
	PROCESSING (one (1) else statement)		/1
	PROCESSING (one (1) loop, either for or while)		/1
	OUTPUT (Neatly displayed)		/1
	Exceptional script (creativity)		/1
9.2	Obtain a value from the user, apply input validation		/1
	For loop correctly implemented		/1
	Display multiplication table		/1
	Determine/Display sum of totals		/1
9.3	BONUS MARKS		
	Obtain the user's email address		
	Implement the loop structure correctly		/2
	Obtain the substrings of each part in variables		
	Display the name and domain		
Comments included in programs (penalty 1 per script), file name according to the specification (penalty 2 per script), correct extension (penalty – not marked), only the latest versions submitted (penalty 1 per additional script submission)			
	TOTAL	**0**	12