# Introduction to Programming

## Lab Worksheet

## Week 2

## Python

Name: Nikita Sah

Level 4 Section: A

British Id: 10011

Level 4 BSc. Hons Computing

Subject: Fundamental Of Computer Programming (FOCP)

The British College (TBC)

## Task:

# Week 2 Practical

# 1. Try inputting the following code and examine the results.

Answer:

```
total = 100
print("The total is", total)
```

# 2. Try inputting the following code and examine the results.

Answer:

```
total = total + 99
print("The total is now", total)
```

# 3. Try inputting the following code, but replace each of the assignment expressions with the equivalent augmented assignment.

Answer:

```
total = total - 1
print("The total is", total)
total = total * 4
print("The total is", total)
total = total / 2
print("The total is", total)
```

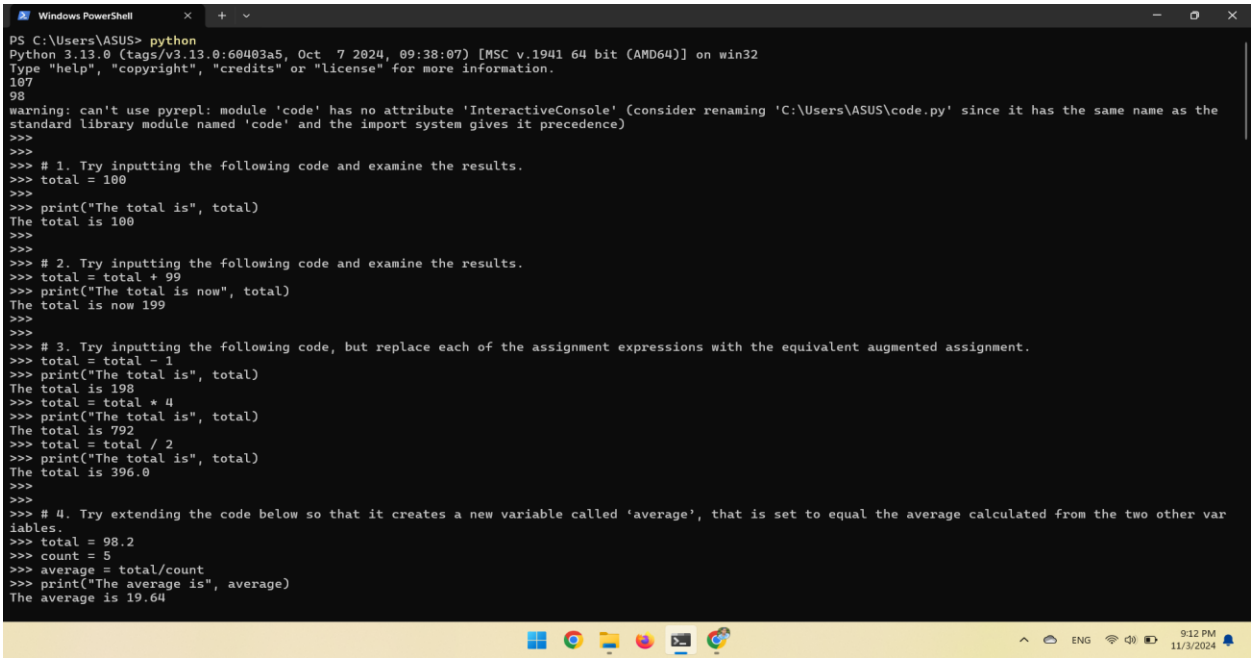
# 4. Try extending the code below so that it creates a new variable called ‘average’, that is set to equal the average calculated from the two other variables.

Answer:

```
total = 98.2
count = 5
```

average = total/count

print("The average is", average)




# 5. Use the type() function to determine the type of each of the following values.

Answer:

type(False)

type(1000)

type(100.111)

type("Hello")

type(True)

type(100 / 5)

type(100 // 5)

# 6. Input the following code and examine the result. What is the * operator doing to the given string operand?

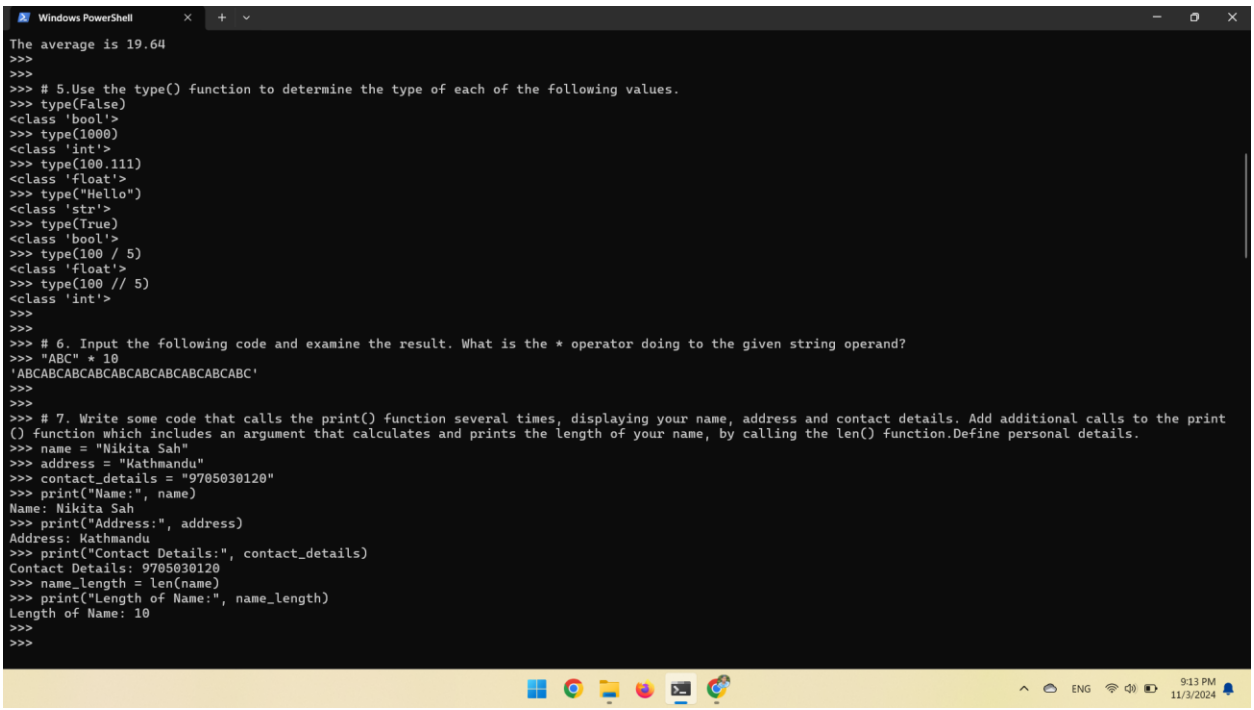
Answer:

"ABC" * 10

# 7. Write some code that calls the print() function several times, displaying your name, address and contact details. Add additional calls to the print() function which includes an argument that calculates and prints the length of your name, by calling the len() function.

Answer:

```
name = "Nikita Sah"

address = "Kathmandu"

contact_details = "9705030120"

# Print personal details

print("Name:", name)

print("Address:", address)

print("Contact Details:", contact_details)

# Calculate and print the length of the name

name_length = len(name)

print("Length of Name:", name_length)
```



# 8. Input the following code, when asked to type your age input a numeric value such as 20. Does this program work? If not, why?

Answer:

```
    age = input("Enter your age ")
    print("in one year your age will be", age + 1)
```

# any value passed to input function has default data type as string,

# Also string and intefer concation is not allowed in python,

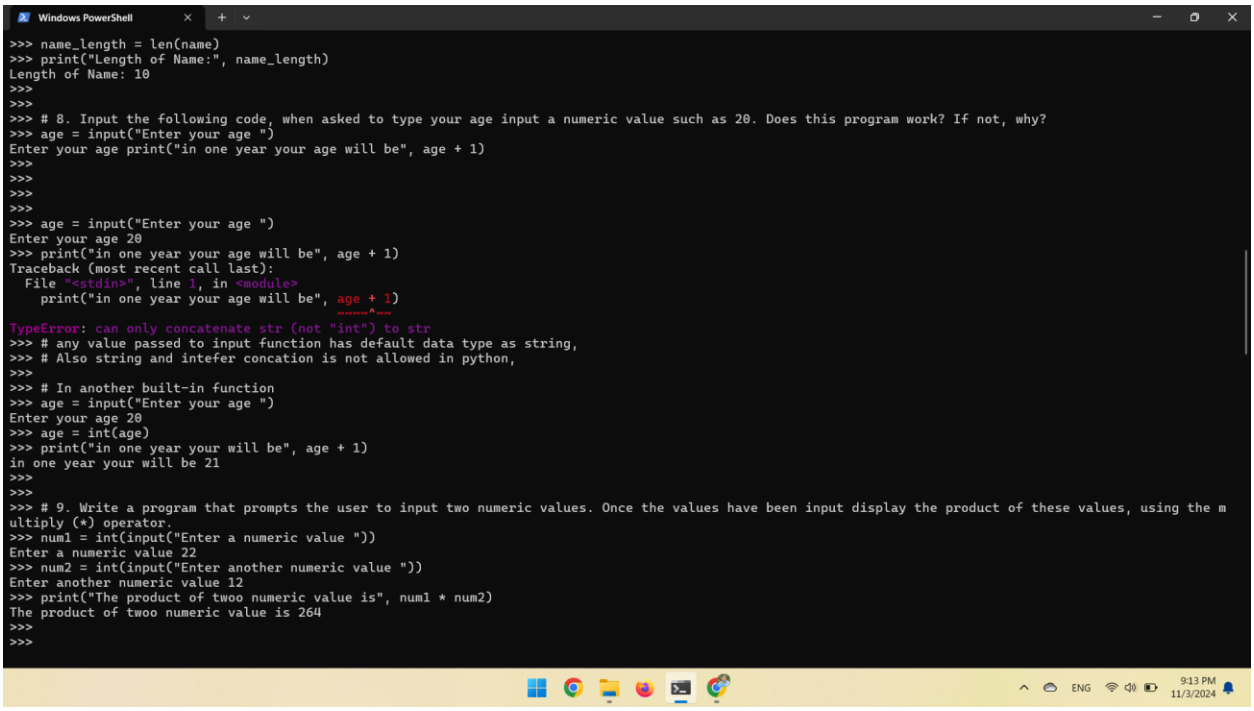
# In another built-in function

```
age = input("Enter your age ")
age = int(age)
print("in one year your will be", age + 1)
```

# 9. Write a program that prompts the user to input two numeric values. Once the values have been input display the product of these values, using the multiply (*) operator.

Answer:

```
    num1 = int(input("Enter a numeric value "))
    num2 = int(input("Enter another numeric value "))
    print("The product of two numeric value is", num1 * num2)
```




# 10. Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown).

"""

This text includes characters such as '\' "" and """,

    This is a new line that starts with a tab

        This new line starts with two tabs

"""

Answer:

```
print("This text includes characters such as '\\' '\"' and \"'\",\n\tThis is a new line that starts with a tab\n\t\tThis new line starts with two tabs")
```

# 11. Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown).

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Hello there!

\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

Answer:

```
print("\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\nHello there!\n\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\")
```

# 12. Write some code that calls a print() function, which takes a single string argument that results in the following text being displayed (exactly as shown). Do this without the use of any escape sequences.

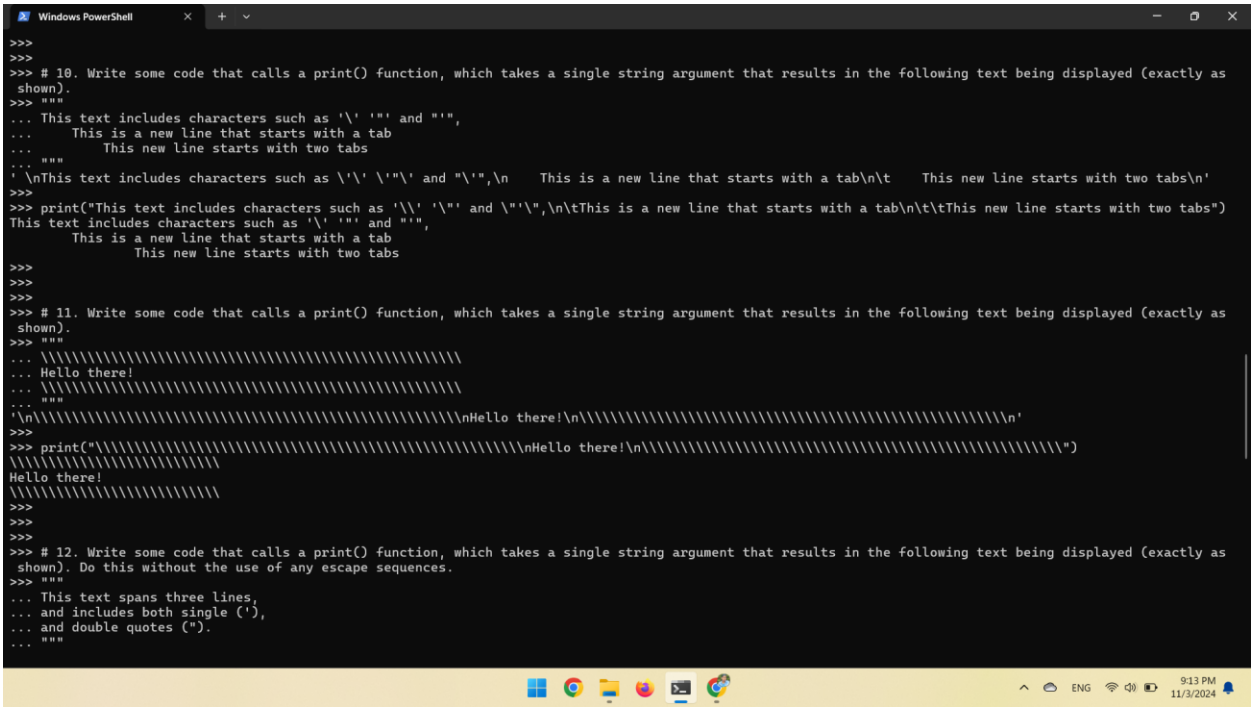
This text spans three lines,

and includes both single ('),

and double quotes (").

Answer:

```
print("""This text spans three lines,

and includes both single ('),

and double quotes (\").""")
```



# 13. Rewrite the above example, so that the third letter of the ‘surname’ is accessed rather than the first, then print this letter to the screen.

Answer:

```
surname = "Palin"
initial = surname[2:3]
print(initial)
```

# 14. Rewrite the above example, so that the tenth letter of the ‘surname’ is accessed, and note the result.

Answer:

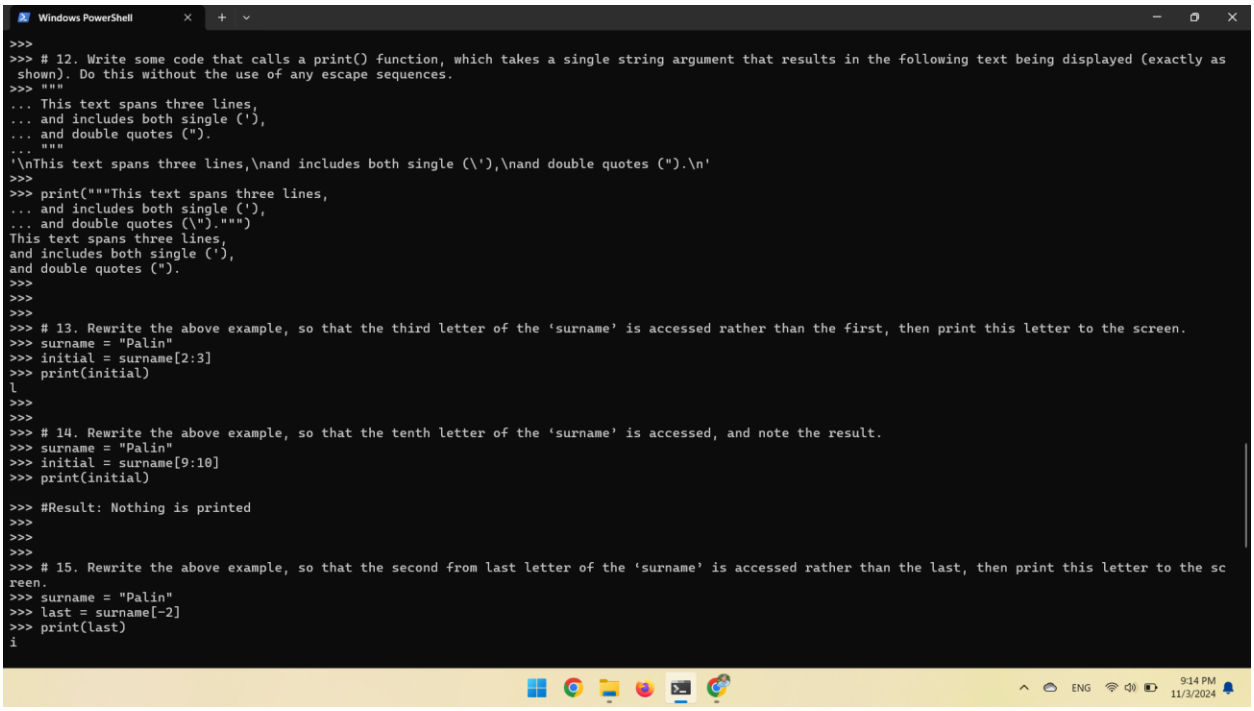
```
surname = "Palin"
initial = surname[9:10]
print(initial)
```

#Result: Nothing is printed

# 15. Rewrite the above example, so that the second from last letter of the ‘surname’ is accessed rather than the last, then print this letter to the screen.

Answer:

```
surname = "Palin"
last = surname[-2]
print(last)
```




# 16. Rewrite the above example, so that all of the characters of the 'surname' except the first character are sliced and then displayed on the screen.

Answer:

```
surname = "Palin"

middle = surname[:1]

print(middle)
```

# 17. Write code that accesses and prints all characters of the 'surname' except the last character.

Answer:

```
surname = "Palin"

first_char_except_last = surname[:-1]

print(first_char_except_last)
```

# 18. Write code that uses slicing to access then print the first four prime numbers defined within the ‘primes’ list given above. Note: you will have to input that list first for testing purposes.

primes = [2, 3, 5, 7, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]

Answer:

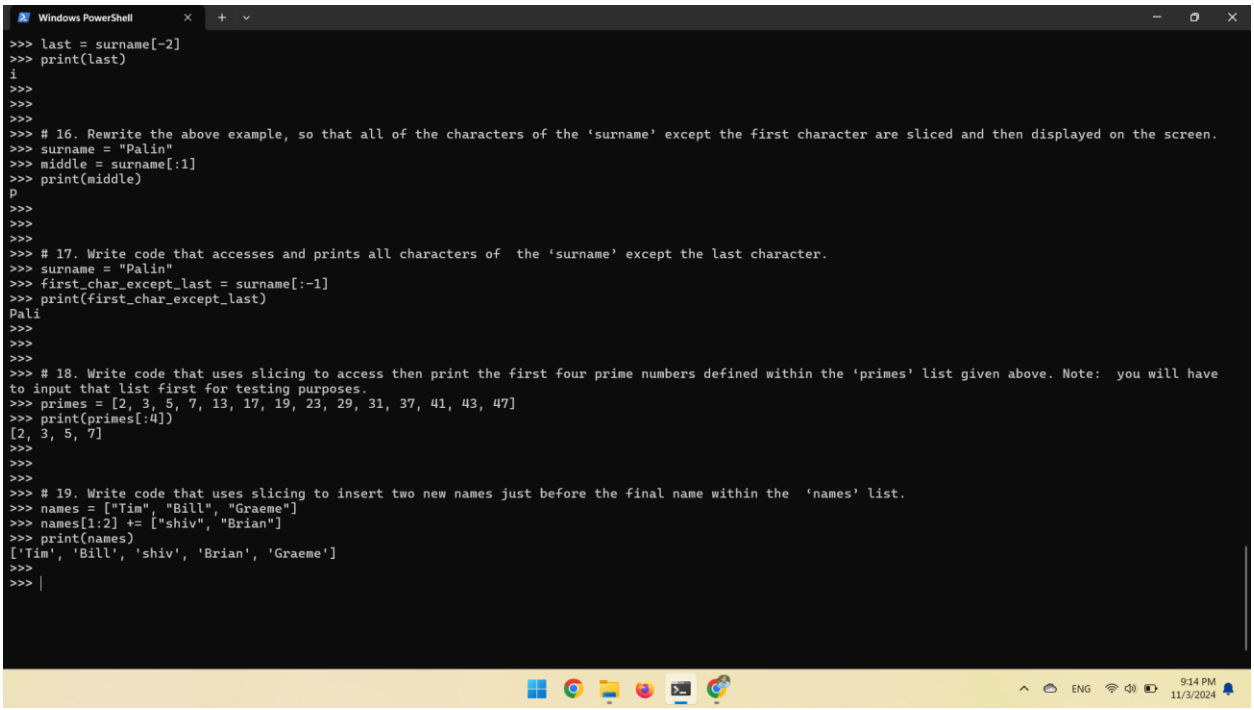
```
print(primes[:4])
```

# 19. Write code that uses slicing to insert two new names just before the final name within the  ‘names’ list.

names = ["Tim", "Bill", "Graeme"]

Answer:

```
names[1:2] += ["shiv", "Brian"]

print(names)
```



# 20. Look at each of the phrases below and ensure you understand what each of these means. For any that you do not understand, do a little research to find a definition of each term. This research may involve looking back over these notes, or the associated lecture notes. It may also involve searching for these terms on the Internet.

# ● Assignment

# ● Data-type

# ● Argument

# ● Indexing

# ● Slicing

# ● Mutable

# ● Immutable

Answer:

- **Assignment:** An assignment is a statement that sets or resets the value of a variable.
- **Data-type:** A data type is an attribute associated with a piece of data that tells a computer system how to interpret its value.
- **Argument:** An argument refers to the value passed to a function.
- **Indexing:** Indexing refers to access of single elements or characters of a list or string.
- **Slicing:** slicing refers to access of multiple elements or characters of a list or string.
- **Mutable:** Mutable is a property of a variable whose value can be changed after it has been created.
- **Immutable:** Immutable is a property of a variable whose value cannot be changed after it has been created.