

# **Fundamentals of Computer Programming (FOCP)**



Submitted by  
Rishav Rauniyar  
Section E  
ID No:8711  
Bsc (Computing)

Submitted to  
Krishna Sir

# Introduction to Programming

## Lab Worksheet

Week 2

## Working with Variables

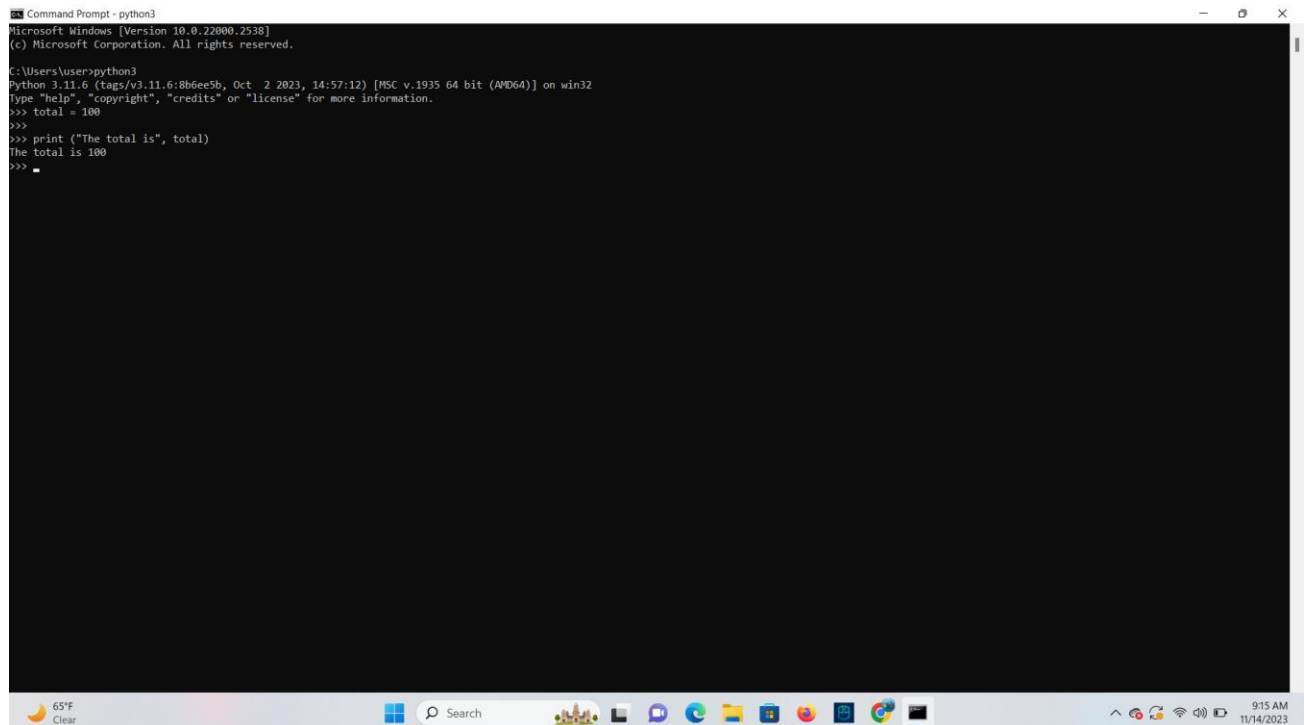
### Variable assignment

**TASK:** Try inputting the following code and examine the results.

```
total = 100
```

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

**Ans**

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the execution of Python code. The output is "The total is 100". The window is set against a black background with white text. The Windows taskbar is visible at the bottom, showing the time as 9:15 AM on 11/14/2023.

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

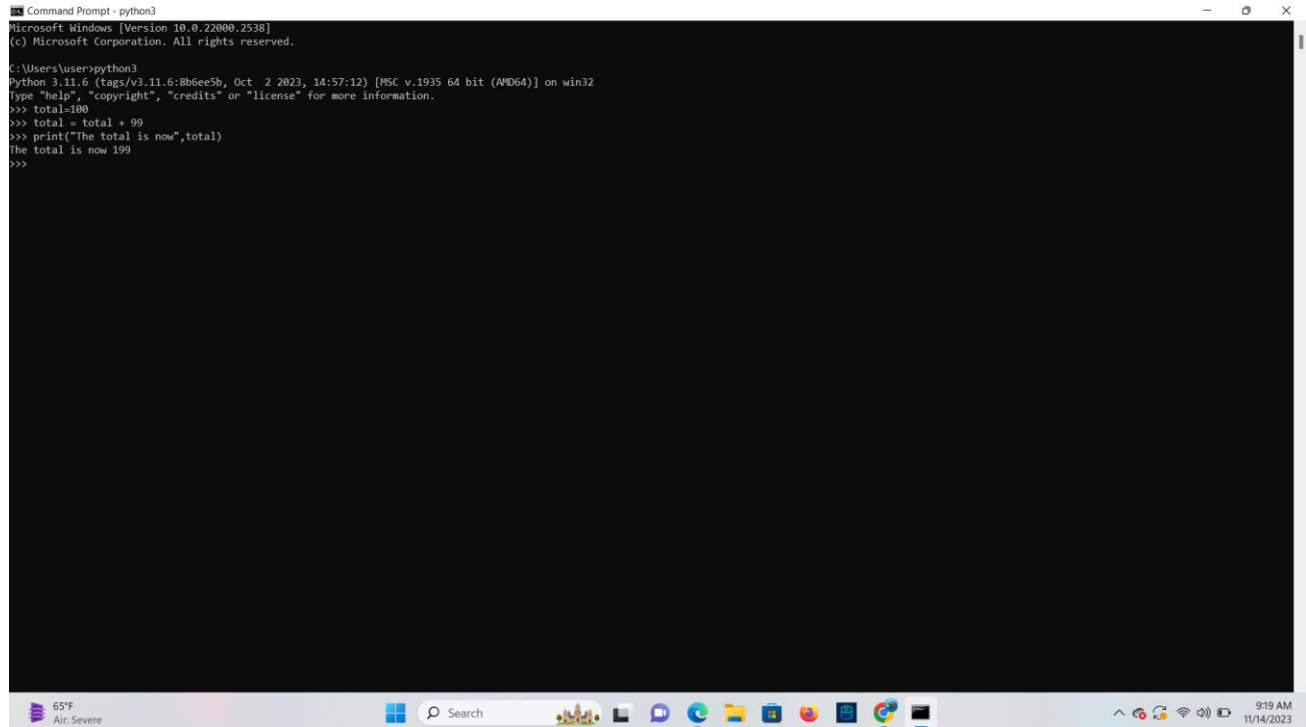
C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> total = 100
>>>
>>> print("The total is", total)
The total is 100
>>>
```

**TASK:** Try inputting the following code and examine the results.

```
total = total + 99
```

```
print("The total is now", total)
```

**Ans**

A screenshot of a Windows Command Prompt window titled "Command Prompt - python3". The window shows the execution of a Python script. The output of the script is displayed in the prompt, showing the variable 'total' being updated and then printed. The Windows taskbar is visible at the bottom of the screen, showing the time as 9:19 AM on 11/14/2023.

```
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> total=100
>>> total = total + 99
>>> print("The total is now",total)
The total is now 199
>>>
```

**TASK:** Try inputting the following code, but replace each of the assignment expressions with the equivalent augmented assignment.

```
total = total - 1
```

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

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> total=100
>>> total=total-1
>>> print("The total is",total)
The total is 99
>>>
```

`total = total * 4`

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

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> total=100
>>> total-total*4
>>> print("The total is",total)
The total is 400
>>> _
```

`total = total / 2`

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

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> total=100
>>> total-total/2
>>> print("The total is",total)
The total is 50.0
>>>
```

**TASK:** 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.

```
total = 98.2
```

```
count = 5
```

```
# add your extra code here
```

**Ans**

```
total = 98.2
```

```
count = 5
```

```
Average = total/count
```

```
Print("The average is:", Average)
```

## Data-Types

### A Variable's data-type

**TASK:** Use the type() function to determine the type of each of the following values.

False

1000

100.111

"Hello"

True

100 / 5

100//5

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> type(False)
<class 'bool'>
>>> type(1000)
<class 'int'>
>>> type(100.111)
<class 'float'>
>>> type("Hello")
<class 'str'>
>>> type(True)
<class 'bool'>
>>> type(100/5)
<class 'float'>
>>> type(100//5)
<class 'int'>
>>>
```

**TASK:** Input the following code and examine the result. What is the \* operator doing to the given string operand?

“ABC” \* 10

**Ans** The \* operator is repeating the string operand 10 times.

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> "ABC"*10
'ABCABCABCABCABCABCABCABC'
>>> _
```

## Calling Built-in functions

**TASK:** 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.

**Ans**

Name= "Rishav"

Address= 123, digital city

Contact= 9844983568



```
print ("My name is:", Name)
print ("My address is:", Address)
print (My Contact is:", Contact)
Name_length=len(Name)
print ("My name length is:", Name_length)
```

## Getting input from the user

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

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

Ans: This program does not work as input() function return a string and when it concatenates with integer 1 it gives us a type error.

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> age = input("Enter your age")
Enter your age21
>>> print("in one year your age will be", age + 1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: can only concatenate str (not "int") to str
>>> _
```

**TASK:** 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.

**Ans**

a=2

b=4

c=a\*b

print ("The product of c is:",c)

## Single, Double and Triple Quotes

comment = 'I would have "thought" you knew better!'

**TASK:** Try writing the above assignment statement but only use double quotes instead of single quotes as the string delimiter. What is the result?

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> comment="I would have "thought" you knew better!"
      File "<stdin>", line 1
        comment="I would have "thought" you knew better!"
                                ^^^^^^^
SyntaxError: invalid syntax
>>>
```

## Escape Sequences

**TASK:** 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

**Ans**

```

C:\Users\user>python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> def display_text():
...     text = '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'
...     print(text)
...
>>> # Call the function to display the desired text
>>> display_text()
This text includes characters such as '\ ' and ' ' '
      This is a new line that starts with a tab
        This new line starts with two tabs
>>>

```

**TASK:** 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!

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> def print_pattern():
...     pattern = r"\\\\"
...     message = "Hello there!"
...
...     print(pattern)
...     print(message)
...     print(pattern)
...
>>> # Call the function to print the desired pattern
>>> print_pattern()
\\\\"
Hello there!
\\\\"
>>> _
```

## Using Triple Quotes

**TASK:** 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 (").

**Ans**

Def print ():

```
    print ("This text spans three lines,\n\n and includes both single ( ' ) ,\n\n and double quotes  
    ( " ).")
```

```
print ()
```

## **Indexing and Slicing**

```
surname = "Palin"
```

```
initial = surname[0]
```

**TASK:** 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.

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> surname = "Palin"
>>> initial = surname[2] # Accessing the third letter (index 2)
>>> print(initial)
l
>>>
```

**TASK:** Rewrite the above example, so that the tenth letter of the ‘surname’ is accessed, and note the result.

**Ans**

```
surname = "Palin"
```

```
tenth_letter = surname[9]
```

```
print("Tenth letter of the surname:", tenth_letter)
```

Output:

Tenth letter of the surname: n

**TASK:** 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.

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> surname = "Palin"
>>> second_to_last_initial = surname[-2] # Access the second-to-last letter
>>> print(second_to_last_initial)
i
>>> _
```



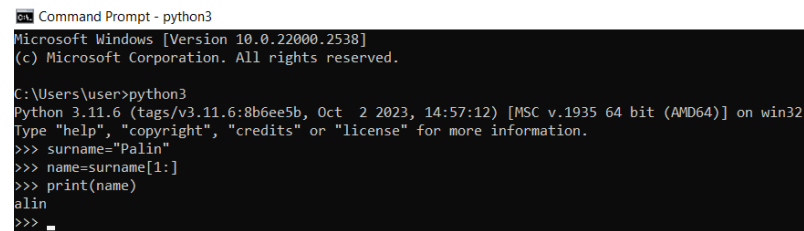
## Slicing

```
surname = "Palin"
```

```
middle = surname[1:4]
```

**TASK:** 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.

## Ans



```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> surname="Palin"
>>> name=surname[1:]
>>> print(name)
alin
>>> _
```

**TASK:** Write code that accesses and prints all characters of the 'surname' except the last

character.

**Ans**

```
Surname=" Rishav"
```

```
for char in surname[:-1]:
```

```
    print(char,end='')
```

## **Introducing Lists**

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

**TASK:** 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.

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> # Input the 'primes' list
>>> primes = [2, 3, 5, 7, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
>>>
>>> # Use slicing to access and print the first four prime numbers
>>> first_four_primes = primes[:4]
>>> print("First four prime numbers:", first_four_primes)
First four prime numbers: [2, 3, 5, 7]
>>> _
```

## Mutable and Immutable types

names = ["Terry", "John", "Michael", "Eric", "Terry", "Graham"]

**TASK:** Write code that uses slicing to insert two new names just before the final name within the 'names' list.

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> names = ["Terry", "John", "Michael", "Eric", "Terry", "Graham"]
>>>
>>> # Two new names to insert
>>> new_names = ["NewName1", "NewName2"]
>>>
>>> # Use slicing to insert the new names before the final name
>>> names[-1:-1] = new_names
>>>
>>> # Print the modified list
>>> print(names)
['Terry', 'John', 'Michael', 'Eric', 'Terry', 'NewName1', 'NewName2', 'Graham']
>>>
```

**TASK:** Work out in your head what the contents of the ‘nums’ list would be, then check this using the Python interpreter.

```
nums = [1,2,3] * 5
```

**Ans**

```
Command Prompt - python3
Microsoft Windows [Version 10.0.22000.2538]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>python3
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> nums = [1,2,3] * 5
>>> print(nums)
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
>>>
```

## Key Terminology

- **Assignment**

**Ans** The operators which is used to assign value to a variable.

- **Data-type**

**Ans** It represents the kind of value that tells what operations can be performed on data.

- **Argument**

**Ans** An argument is a value that is passed to a function when it is called.

- **Indexing**

**Ans** Indexing is the process of accessing an element in a sequence.

- **Slicing**

**Ans** It enables users to access the specific range of elements by mentioning their indices.

- **Mutable**

**Ans** Those who values can be modified once they are created.

- **Immutable**

**Ans** Those who values cannot be modified once they are created.