



**FEU INSTITUTE OF TECHNOLOGY**

**COLLEGE OF COMPUTER STUDIES**

**IT0011**

**Integrative Programming and  
Technologies**

**EXERCISE**

---

**3**

**String and File Handling**

<b>Student Name:</b>	CUETO, ALEXA JOYCE G.
<b>Section:</b>	TW23
<b>Professor:</b>	Sir Joseph Calleja

## **I. PROGRAM OUTCOME (PO) ADDRESSED**

Analyze a complex problem and identify and define the computing requirements appropriate to its solution.

## **II. LEARNING OUTCOME (LO) ADDRESSED**

Utilize string manipulation techniques and file handling in Python

## **III. INTENDED LEARNING OUTCOMES (ILO)**

At the end of this exercise, students must be able to:

- Perform common string manipulations, such as concatenation, slicing, and formatting.
- Understand and use file handling techniques to read from and write to files in Python.
- Apply string manipulation and file handling to solve practical programming problems.

## **IV. BACKGROUND INFORMATION**

### **String Manipulation:**

String manipulation is a crucial aspect of programming that involves modifying and processing textual data. In Python, strings are versatile, and several operations can be performed on them. This exercise focuses on fundamental string manipulations, including concatenation (combining strings), slicing (extracting portions of strings), and formatting (constructing dynamic strings).

Common String Methods:

- `len()`: Returns the length of a string.
- `lower()`, `upper()`: Convert a string to lowercase or uppercase.
- `replace()`: Replace a specified substring with another.
- `count()`: Count the occurrences of a substring within a string.

### **File Handling:**

File handling is essential for reading and writing data to external files, providing a way to store and retrieve information. Python offers straightforward mechanisms for file manipulation. This exercise introduces the basics of file handling, covering the opening and closing of files, as well as reading from and writing to text files.

Understanding File Modes:

- `'r'` (read): Opens a file for reading.
- `'w'` (write): Opens a file for writing, overwriting the file if it exists.
- `'a'` (append): Opens a file for writing, appending to the end of the file if it exists.

Understanding string manipulation and file handling is fundamental for processing and managing data in Python programs. String manipulations allow for the transformation and extraction of information from textual data, while file handling enables interaction with external data sources. Both skills are essential for developing practical applications and solving real-world programming challenges. The exercises in this session aim to reinforce these concepts through hands-on practice and problem-solving scenarios.

## V. GRADING SYSTEM / RUBRIC

Criteria	Excellent (5)	Good (4)	Satisfactory (3)	Needs Improvement (2)	Unsatisfactory (1)
<b>Correctness</b>	Code functions correctly and meets all requirements.	Code mostly functions as expected and meets most requirements.	Code partially functions but may have logical errors or missing requirements.	Code has significant errors, preventing proper execution.	Code is incomplete or not functioning.
<b>Code Structure</b>	Code is well-organized with clear structure and proper use of functions.	Code is mostly organized with some room for improvement in structure and readability.	Code lacks organization, making it somewhat difficult to follow.	Code structure is chaotic, making it challenging to understand.	Code lacks basic organization.
<b>Documentation</b>	Comprehensive comments and docstrings provide clarity on the code's purpose.	Sufficient comments and docstrings aid understanding but may lack details in some areas.	Limited comments, making it somewhat challenging to understand the code.	Minimal documentation, leaving significant gaps in understanding.	No comments or documentation provided.
<b>Coding Style</b>	Adheres to basic coding style guidelines, with consistent and clean practices.	Mostly follows coding style guidelines, with a few style inconsistencies.	Style deviations are noticeable, impacting code readability.	Significant style issues, making the code difficult to read.	No attention to coding style; the code is messy and unreadable.
<b>Effort and Creativity</b>	Demonstrates a high level of effort and creativity, going beyond basic requirements.	Shows effort and creativity in addressing most requirements.	Adequate effort but lacks creativity or exploration beyond the basics.	Minimal effort and creativity evident.	Little to no effort or creativity apparent.

## VI. LABORATORY ACTIVITY

### INSTRUCTIONS:

Copy your source codes to be pasted in this document as well as a screen shot of your running output.

### 3.1. Activity for Performing String Manipulations

Objective: To perform common and practical string manipulations in Python.

Task: Write a Python program that includes the following string manipulations:

- Concatenate your first name and last name into a full name.
- Slice the full name to extract the first three characters of the first name.
- Use string formatting to create a greeting message that includes the sliced first name

Sample Output

```
Enter your first name: Peter
Enter your last name: Parker
Enter your age: 20

Full Name: Peter Parker
Sliced Name: Pete
Greeting Message: Hello, Pete! Welcome. You are 20 years old.
```

My Source Code:

*#3.1*

*#CUETO, ALEXA JOYCE G.*

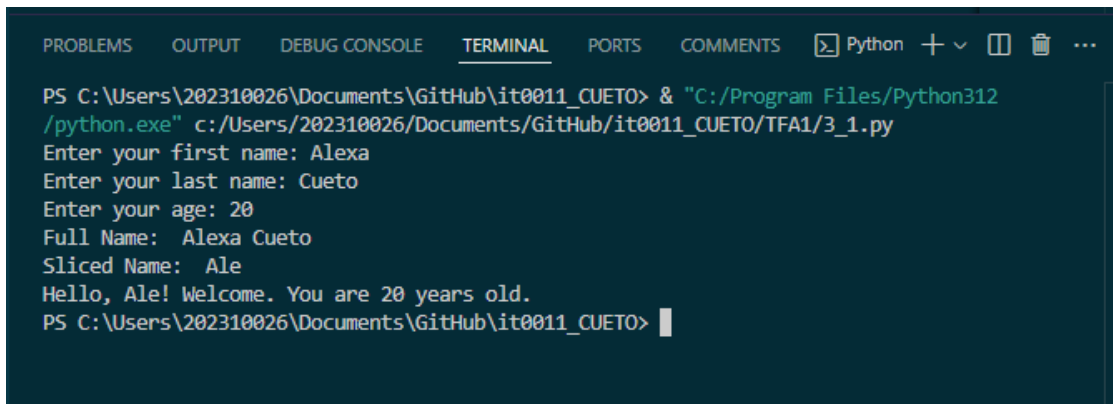
*#TW23*

```
fName = input("Enter your first name: ")
lName = input("Enter your last name: ")
age = int(input("Enter your age: "))

slicedFName = fName[0:3]
formattedString = f"Hello, {slicedFName}! Welcome. You are {age}
years old. "

print("Full Name: ", fName + " " + lName)
print("Sliced Name: ", slicedFName)
print(formattedString)
```

My Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + v [] [X] ...
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> & "C:/Program Files/Python312
/python.exe" c:/Users/202310026/Documents/GitHub/it0011_CUETO/TFA1/3_1.py
Enter your first name: Alexa
Enter your last name: Cueto
Enter your age: 20
Full Name:  Alexa Cueto
Sliced Name:  Ale
Hello, Ale! Welcome. You are 20 years old.
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> |
```

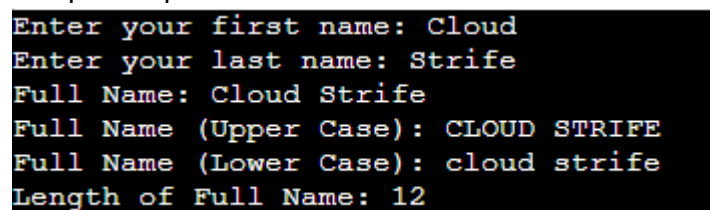
### 3.2 Activity for Performing String Manipulations

Objective: To perform common and practical string manipulations in Python.

Task: Write a Python program that includes the following string manipulations:

- Input the user's first name and last name.
- Concatenate the input names into a full name.
- Display the full name in both upper and lower case.
- Count and display the length of the full name

Sample Output



```
Enter your first name: Cloud
Enter your last name: Strife
Full Name: Cloud Strife
Full Name (Upper Case): CLOUD STRIFE
Full Name (Lower Case): cloud strife
Length of Full Name: 12
```

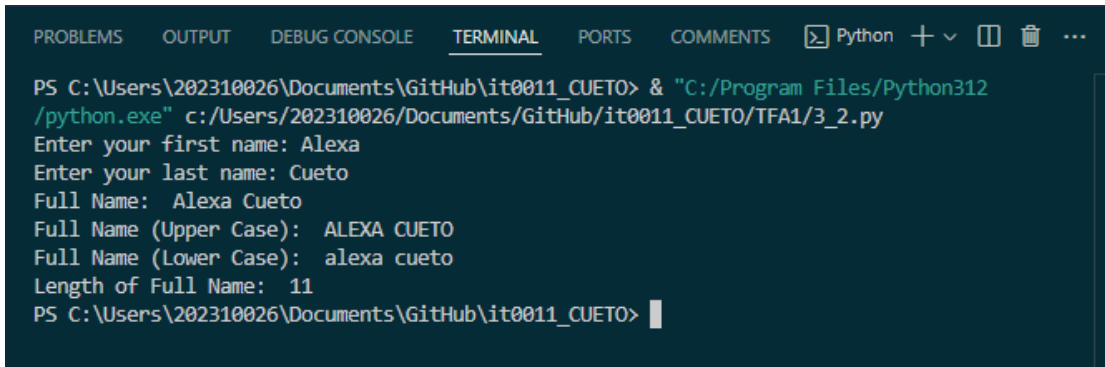
My Source Code:

```
#3_2
#CUETO, ALEXA JOYCE G.
#TW23

fName = input("Enter your first name: ")
lName = input("Enter your last name: ")
fullName = fName + " " + lName
length = len(fullName)
```

```
print("Full Name: ", fullName)
print("Full Name (Upper Case): ", fullName.upper())
print("Full Name (Lower Case): ", fullName.lower())
print("Length of Full Name: ", length)
```

My Output:



```
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> & "C:/Program Files/Python312/python.exe" c:/Users/202310026/Documents/GitHub/it0011_CUETO/TFA1/3_2.py
Enter your first name: Alexa
Enter your last name: Cueto
Full Name: Alexa Cueto
Full Name (Upper Case): ALEXA CUETO
Full Name (Lower Case): alexa cueto
Length of Full Name: 11
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> |
```

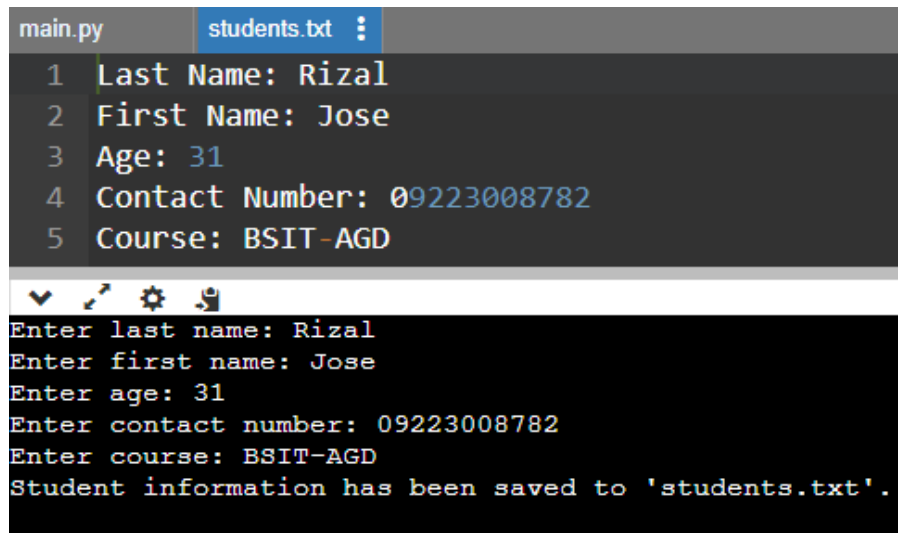
### 3.3. Practical Problem Solving with String Manipulation and File Handling

Objective: Apply string manipulation and file handling techniques to store student information in a file.

Task: Write a Python program that does the following:

- Accepts input for the last name, first name, age, contact number, and course from the user.
- Creates a string containing the collected information in a formatted way.
- Opens a file named "students.txt" in append mode and writes the formatted information to the file.
- Displays a confirmation message indicating that the information has been saved.

Sample Output



The screenshot shows a code editor with two tabs: 'main.py' and 'students.txt'. The 'main.py' tab is active, displaying a Python script with five lines of code. Below the code editor is a terminal window showing the execution of the script. The terminal output shows the user entering the values for each variable, and a final message indicating that the student information has been saved to 'students.txt'.

```
main.py students.txt :
1 Last Name: Rizal
2 First Name: Jose
3 Age: 31
4 Contact Number: 09223008782
5 Course: BSIT-AGD

Enter last name: Rizal
Enter first name: Jose
Enter age: 31
Enter contact number: 09223008782
Enter course: BSIT-AGD
Student information has been saved to 'students.txt'.
```

My Source Code:

```
#3_3
#CUETO, ALEXA JOYCE G.
#TW23

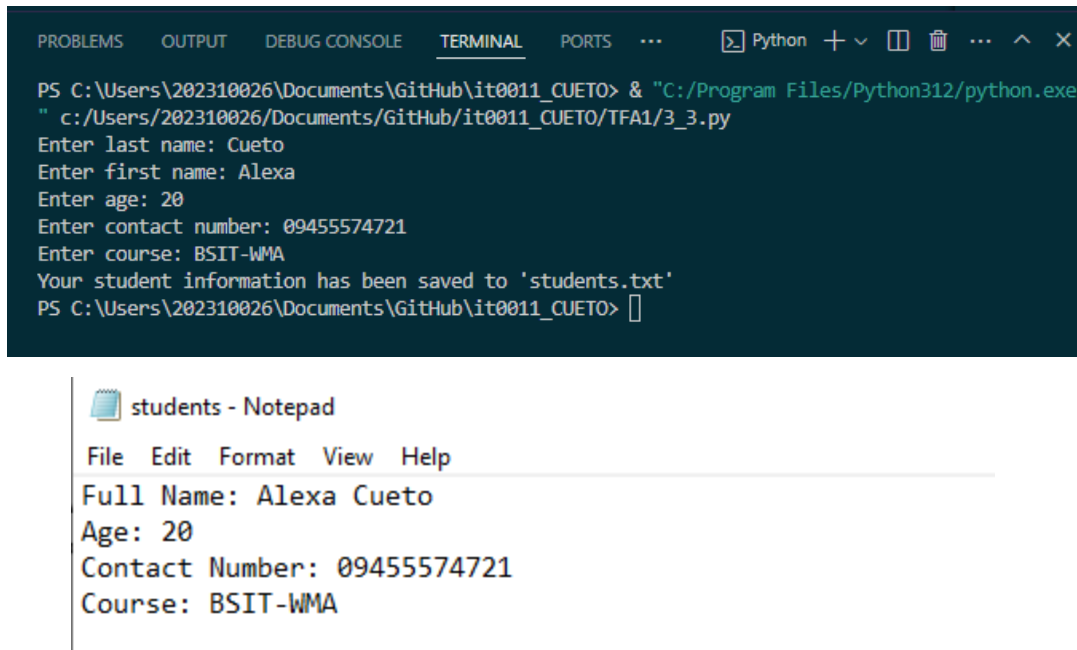
lName = input("Enter last name: ")
fName = input("Enter first name: ")
age = int(input("Enter age: "))
contactNo = input("Enter contact number: ")
course = input("Enter course: ")

infoString = f"Full Name: {fName} {lName}\nAge: {age}\nContact
Number: {contactNo}\nCourse: {course}"

file1 = open("students.txt", "w")
W = (infoString)
file1.writelines(W)
file1.close()

print("Your student information has been saved to
'students.txt'")
```

My Output:



The image shows two windows. The top window is a Python terminal with the following text:

```
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> & "C:/Program Files/Python312/python.exe"
" c:/Users/202310026/Documents/GitHub/it0011_CUETO/TFA1/3_3.py
Enter last name: Cueto
Enter first name: Alexa
Enter age: 20
Enter contact number: 09455574721
Enter course: BSIT-WMA
Your student information has been saved to 'students.txt'
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> 
```

The bottom window is a Notepad application titled "students - Notepad" with the following text:

```
File Edit Format View Help
Full Name: Alexa Cueto
Age: 20
Contact Number: 09455574721
Course: BSIT-WMA
```

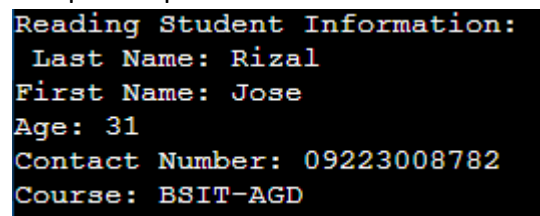
### 3.4 Activity for Reading File Contents and Display

Objective: Apply file handling techniques to read and display student information from a file.

Task: Write a Python program that does the following:

- Opens the "students.txt" file in read mode.
- Reads the contents of the file.
- Displays the student information to the user

Sample Output



```
Reading Student Information:
Last Name: Rizal
First Name: Jose
Age: 31
Contact Number: 09223008782
Course: BSIT-AGD
```



### My Source Code:

```
#3_4
#CUETO, ALEXA JOYCE G.
#TW23

file1 = open("students.txt", "r")
print("Reading Student Information: ")
print(file1.read())
print()
file1.close
```

### My Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS Python + v []
PS C:\Users\202310026\Documents\GitHub\it0011_CUETO> & "C:/Program Files/Python312/p
ython.exe" c:/Users/202310026/Documents/GitHub/it0011_CUETO/TFA2/3_4.py
Reading Student Information:
Full Name: Alexa Cueto
Age: 20
Contact Number: 09455574721
Course: BSIT-WMA

PS C:\Users\202310026\Documents\GitHub\it0011_CUETO>
```

### QUESTION AND ANSWER:

1. How does the format() function help in combining variables with text in Python? Can you provide a simple example?

-With the format() function, it enables the programmers to structure and format a string with variables in a more flexible and easier way as programmers can control the way the output looks. The placeholders represented by {} concatenates the text and values to be printed which makes the code more coherent and readable. An example of its syntax is:

```
age = 10
print("My age is {age});
```

2. Explain the basic difference between opening a file in 'read' mode ('r') and 'write' mode ('w') in Python. When would you use each

-The read mode opens an existing file and reads the information included inside. On the other hand, write mode opens or creates (if a file is not existing) a file for writing where it stores new information or text. Read mode is used when the programmer wants to simply print or read an existing file, while write mode is used when a programmer wants to override the existing contents through writing a new information.

3. Describe what string slicing is in Python. Provide a basic example of extracting a substring from a larger string.

-String slicing allows programmers to remove or extract specific letters in a word using index range. The syntax follows a string[starting index : ending index]. An example of this is: slicedAddress = address[0:5] which gets letters starting from the 0 index up to the 5<sup>th</sup> index.

4. When saving information to a file in Python, what is the purpose of using the 'a' mode instead of the 'w' mode? Provide a straightforward example.

-The 'a' mode which is append, opens a file for writing but it does not overwrite the existing information or content included in a file. It simply adds the new content. This is useful for audit logs where previous data are important.

5. Write a simple Python code snippet to open and read a file named "data.txt." How would you handle the case where the file might not exist?

try:

```
file1 = open("data.txt", "r")
print(file1.read())
print()
file1.close()
```

except FileNotFoundError:

```
print("No existing files.")
```