COLLEGE OF COMPUTER STUDIES

IT0011
Integrative Programming and Technologies

EXERCISE

3

String and File Handling

Student Name:	Aries Ong
Section:	TB22
Professor:	Sir. 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)
Correctness	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.
Code Structure	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.
Documentation	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.
Coding Style	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.
Effort and Creativity	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.
```
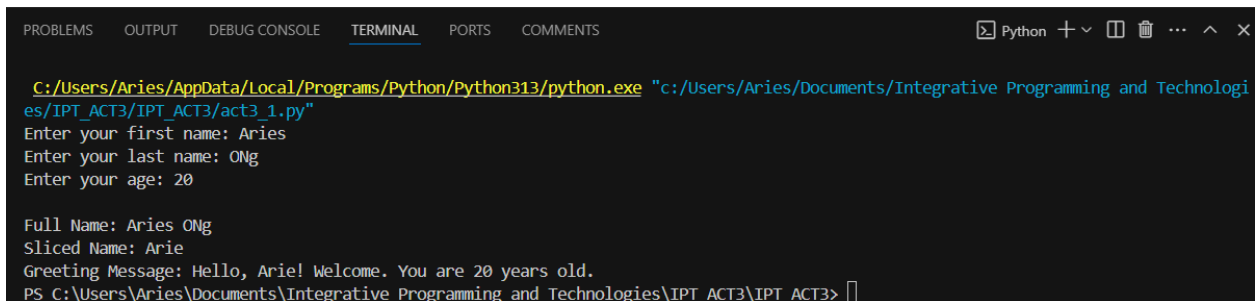
Code:
```python
# Asking the user for their first name, last name, and age
first = input("Enter your first name: ")
last = input("Enter your last name: ")
age = input("Enter your age: ")

# Combining the first and last name to create the full name
full = first + " " + last

# Extracting the first four characters of the first name
sliced = first[:4]

# Displaying the full name, sliced name, and a greeting message
print("\nFull Name:", full)
print("Sliced Name:", sliced)
print(f"Greeting Message: Hello, {sliced}! Welcome. You are {age} years old.")
```

Output:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   COMMENTS                     Python + ∨ □ 🗑 ··· ∧ ×

 C:/Users/Aries/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/Aries/Documents/Integrative Programming and Technologi
 es/IPT_ACT3/IPT_ACT3/act3_1.py"
 Enter your first name: Aries
 Enter your last name: ONg
 Enter your age: 20

 Full Name: Aries ONg
 Sliced Name: Arie
 Greeting Message: Hello, Arie! Welcome. You are 20 years old.
 PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3> |
```

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
```

Code:
```python
# Asking the user for their first name, last name, and age
first = input("Enter your first name: ")
last = input("Enter your last name: ")

# Combining first and last name
full_name = first + " " + last

# Changing the full name to upper and lower case as well as getting the length
upper = full_name.upper()
lower = full_name.lower()
length = len(full_name)

# Displaying the full name, full name in upper case, full name in lower case, and the length of
the full name
print("\nFull Name:", full_name)
print("Full Name (Upper Case):", upper)
print("Full Name (Lower Case):", lower)
print("Length of Full Name:", length)
```

Output:
```
cal/Programs/Python/Python313/python.exe "c:/Users/Aries/Documents/Integrative Programming and Technologies/IPT_ACT
3/IPT_ACT3/act3_2.py"
Enter your first name: Aries
Enter your last name: Ong

Full Name: Aries Ong
Full Name (Upper Case): ARIES ONG
Full Name (Lower Case): aries ong
Length of Full Name: 9
PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3>
```
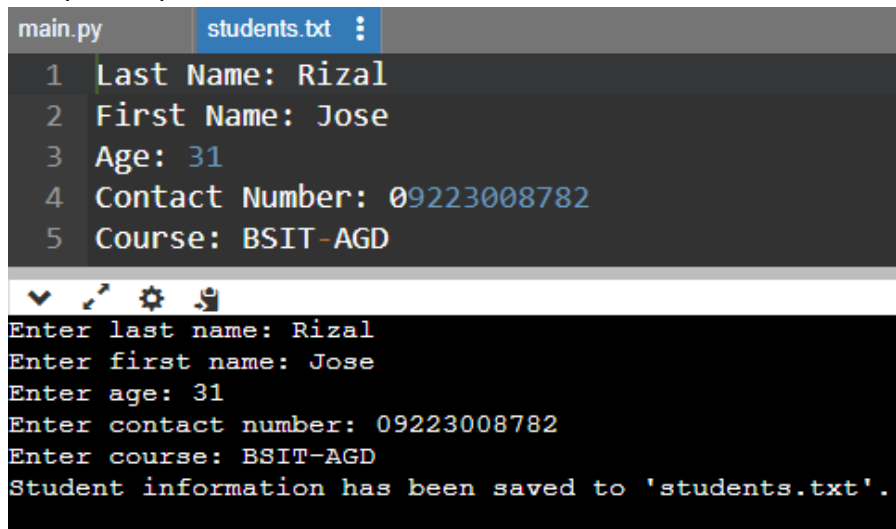
3.3. Practical Problem Solving with String Manipulation and File Handling
Objective: Apply string manipulation and file handling techniques to store student information in
a file.

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

```
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'.
```

Code:
```python
# Asking user for their last name, first name, age, contact number, and course
last = input("Enter your last name: ")
first = input("Enter your first name: ")
age = input("Enter age: ")
contact = input("Enter contact number: ")
course = input("Enter course: ")

# Formatting the collected information
student_info = f"Last Name: {last}\nFirst Name: {first}\nAge: {age}\nContact Number:
{contact}\nCourse: {course}\n\n"

# Writing to a file in append mode
with open("students.txt", "a") as file:
    file.write(student_info)

# Display confirmation message
print("Student information has been saved to 'students.txt'.")
```
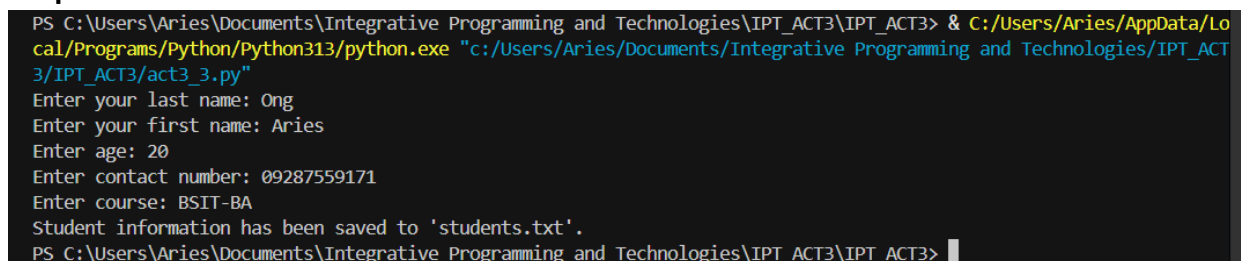
Output:
```
PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3> & C:/Users/Aries/AppData/Lo
cal/Programs/Python/Python313/python.exe "c:/Users/Aries/Documents/Integrative Programming and Technologies/IPT_ACT
3/IPT_ACT3/act3_3.py"
Enter your last name: Ong
Enter your first name: Aries
Enter age: 20
Enter contact number: 09287559171
Enter course: BSIT-BA
Student information has been saved to 'students.txt'.
PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3>
```

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
```

Code:
```python
# Define the file name that contains the student information
filename = "students.txt"

try:
    # Code to try to open the file in read mode
    with open(filename, "r") as file:
        print("Student Information:\n")

        # if successful, read and print the content of the file
        print(file.read())
except FileNotFoundError:
    # A error handler to print a message if the file is not found
    print(f"Error: '{filename}' not found. Please add student information first.")
```

Output:
```
PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3> & C:/Users/Aries/AppData/Lo
cal/Programs/Python/Python313/python.exe "c:/Users/Aries/Documents/Integrative Programming and Technologies/IPT_ACT
3/IPT_ACT3/act3_4.py"
Student Information:

Last Name: Ong
First Name: Aries
Age: 20
Contact Number: 09287559171
Course: BSIT-BA

PS C:\Users\Aries\Documents\Integrative Programming and Technologies\IPT_ACT3\IPT_ACT3>
```

QUESTION AND ANSWER:

1. How does the format() function help in combining variables with text in Python? Can you provide a simple example?
The format() function can help with clarity of the code for example in activity 1 i used it to print the greeting message as shown below.
```python
# Displaying the full name, sliced name, and a greeting message
print("\nFull Name:", full)
print("Sliced Name:", sliced)
print("Greeting Message: Hello, {}! Welcome. You are {} years old.".format(sliced, 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 is used to view a content of txt file such as done in activity 4 and write mode is used to place data into a txt file which was used in activity 3.

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

This is used in the first activity where sliced = first[:4] is used to get the first 4 characters of the word or the value stored in the variable first. The number 4 inside the brackets are the range or until what index or character is shown. You can also use negative number to reverse the order going right to left or starting form the last letter/ character of the word.

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.

Appends or 'a' mode is to add new content to a file without removing existing data. While write or 'w' mode overwrites the existing context to add the new inputted content.

```python
# Writing to a file in append mode
with open("students.txt", "a") as file:
    file.write(student_info)

# Display confirmation message
print("Student information has been saved to 'students.txt'.")
```

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?

```python
except FileNotFoundError:
    # A error handler to print a message if the file is not found
    print(f"Error: '{filename}' not found. Please add student information first.")
```