Python Task 5

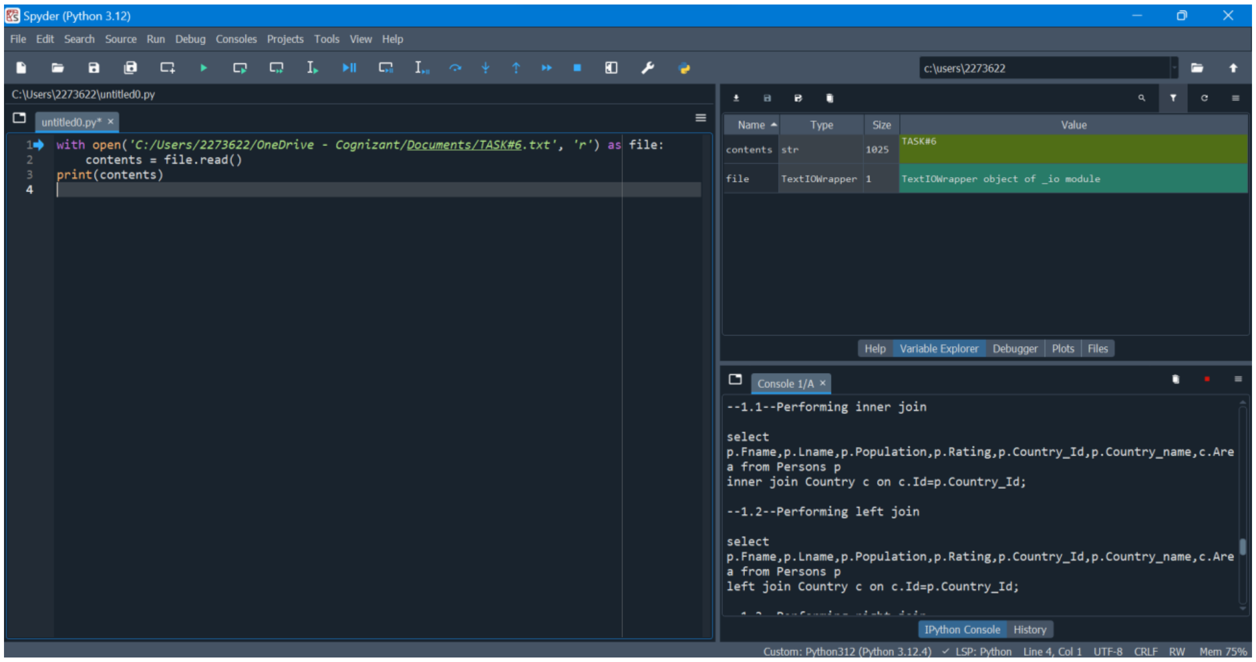
Exercise 1:
Write a Python program to read a file and display its contents.

Definition
The provided Python program is designed to read the contents of a file and display them. It uses the open function to access the file and the read method to read its contents.

Purpose
The primary purpose of this program is to:
1. **Access a file**: Open a file in read mode.
2. **Read the file's contents**: Retrieve the data stored in the file.
3. **Display the contents**: Output the data to the console.

Use Cases
This type of program can be useful in various scenarios, such as:
1. **Log File Analysis**: Reading and displaying log files to monitor application behavior or debug issues.
2. **Configuration Files**: Loading and displaying configuration settings from a file.
3. **Data Processing**: Reading data files for further processing or analysis.
4. **Educational Purposes**: Teaching basic file handling in Python.

Code:
```
with open('C:/Users/2273622/OneDrive - Cognizant/Documents/TASK#6.txt', 'r') as file:
    contents = file.read()
print(contents)
```



Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\2273622\untitled0.py
untitled0.py*
c:\users\2273622
Help Variable Explorer Debugger Plots Files
Console 1/A
IPython Console History
Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 4, Col 1 UTF-8 CRLF RW Mem 75%

```
with open('C:/Users/2273622/OneDrive - Cognizant/Documents/TASK#6.txt', 'r') as file:
    contents = file.read()
print(contents)
```

Name	Type	Size	Value
contents	str	1025	TASK#6
file	TextIOWrapper	1	TextIOWrapper object of _io module

```
--1.1--Performing inner join

select
p.Fname,p.Lname,p.Population,p.Rating,p.Country_Id,p.Country_name,c.Are
a from Persons p
inner join Country c on c.Id=p.Country_Id;

--1.2--Performing left join

select
p.Fname,p.Lname,p.Population,p.Rating,p.Country_Id,p.Country_name,c.Are
a from Persons p
left join Country c on c.Id=p.Country_Id;
```

Exercise 2:
Write a Python program to copy the contents of one file to another file.

Definition
The provided Python program is designed to copy the contents of one file and write them to another file. It uses the open function to read from the source file and write to the destination file.

Purpose
The primary purpose of this program is to:

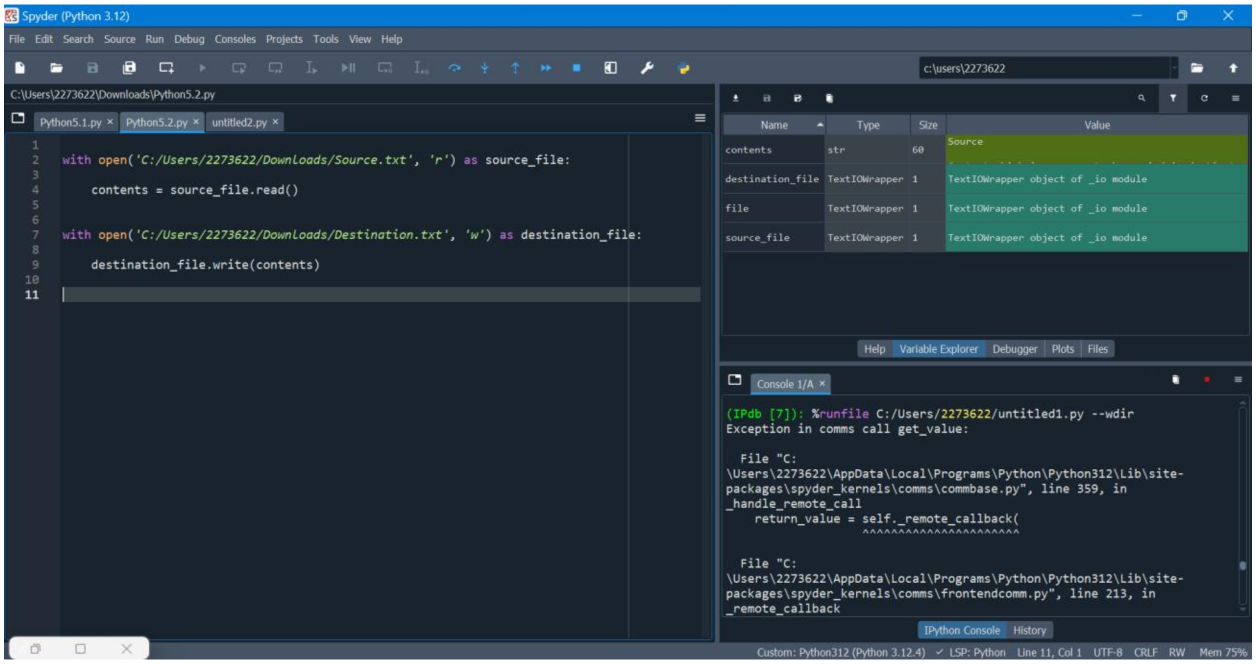
1. **Read data from a source file**: Open and read the contents of the source file.
2. **Write data to a destination file**: Open the destination file and write the read contents into it.

Use Cases
This type of program can be useful in various scenarios, such as:

1. **File Backup**: Creating a backup copy of important files.
2. **Data Migration**: Moving data from one file to another.
3. **File Duplication**: Creating duplicates of files for testing or distribution.
4. **Log Archiving**: Copying log files to an archive location for long-term storage.

Code:

```
with open('C:/Users/2273622/Downloads/Source.txt', 'r') as source_file:
    contents = source_file.read()

with open('C:/Users/2273622/Downloads/Destination.txt', 'w') as destination_file:
    destination_file.write(contents)
```



C:\Users\2273622\Downloads\Python5.2.py

```python

with open('C:/Users/2273622/Downloads/Source.txt', 'r') as source_file:

    contents = source_file.read()


with open('C:/Users/2273622/Downloads/Destination.txt', 'w') as destination_file:

    destination_file.write(contents)

```

Name	Type	Size	Value
contents	str	60	Source
destination_file	TextIOWrapper	1	TextIOWrapper object of _io module
file	TextIOWrapper	1	TextIOWrapper object of _io module
source_file	TextIOWrapper	1	TextIOWrapper object of _io module

```
(IPdb [7]): %runfile C:/Users/2273622/untitled1.py --wdir
Exception in comms call get_value:

  File "C:
\Users\2273622\AppData\Local\Programs\Python\Python312\Lib\site-
packages\spyder_kernels\comms\commbase.py", line 359, in
_handle_remote_call
    return_value = self._remote_callback(
                   ^^^^^^^^^^^^^^^^^^^^^^


  File "C:
\Users\2273622\AppData\Local\Programs\Python\Python312\Lib\site-
packages\spyder_kernels\comms\frontendcomm.py", line 213, in
_remote_callback
```

Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 11, Col 1 UTF-8 CRLF RW Mem 75%

Exercise 3:
Write a Python program to read the content of a file and count the total number of words in that file.

Definition

This Python program reads the content of a specified file, splits the content into individual words, and counts the total number of words.

Purpose

The primary purpose of this program is to:

1. **Read file content**: Open and read the entire content of a file.
2. **Process text data**: Split the text into words.
3. **Count words**: Calculate the total number of words in the file.

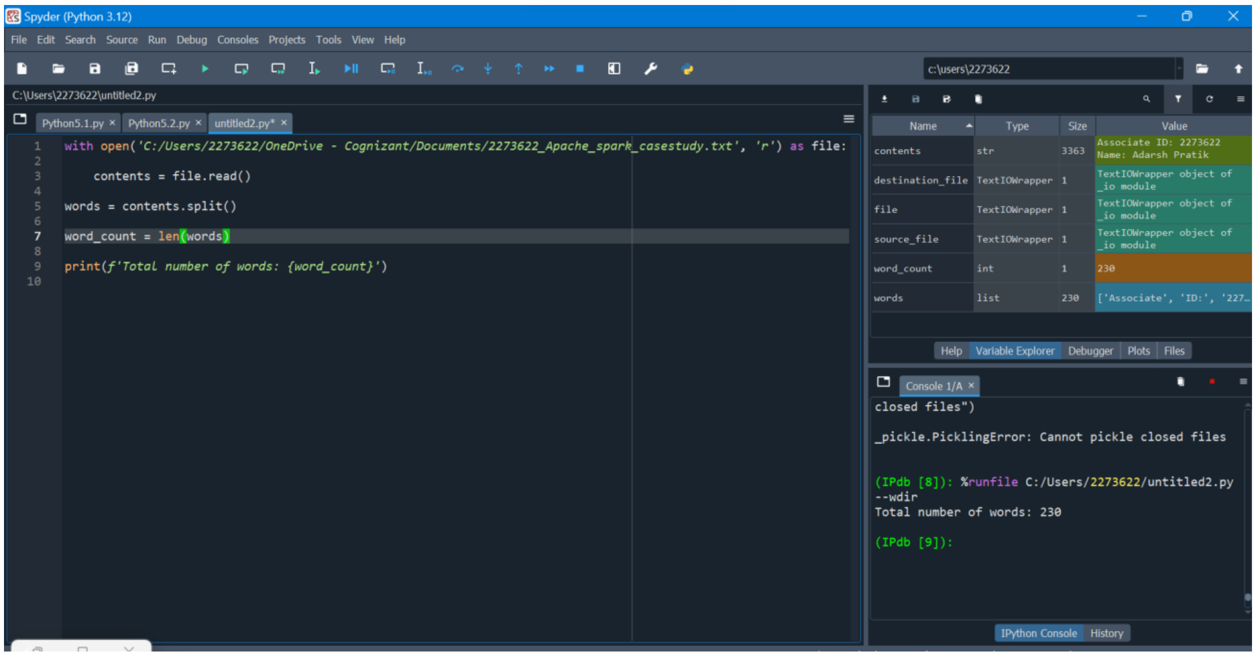
Use Cases

This type of program can be useful in various scenarios, such as:

1. **Text Analysis**: Analyzing the length and complexity of documents by counting words.
2. **Data Processing**: Preparing text data for further processing or analysis.
3. **Content Management**: Checking the word count of articles, essays, or reports.

Code:

```python
with open('C:/Users/2273622/OneDrive -
Cognizant/Documents/2273622_Apache_spark_casestudy.txt', 'r') as file:
    contents = file.read()
words = contents.split()
word_count = len(words)
print(f'Total number of words: {word_count}')
```



```python
with open('C:/Users/2273622/OneDrive - Cognizant/Documents/2273622_Apache_spark_casestudy.txt', 'r') as file:

    contents = file.read()

words = contents.split()

word_count = len(words)

print(f'Total number of words: {word_count}')
```

Name	Type	Size	Value
contents	str	3363	Associate ID: 2273622 Name: Adarsh Pratik
destination_file	TextIOWrapper	1	TextIOWrapper object of _io module
file	TextIOWrapper	1	TextIOWrapper object of _io module
source_file	TextIOWrapper	1	TextIOWrapper object of _io module
word_count	int	1	230
words	list	230	['Associate', 'ID:', '227…

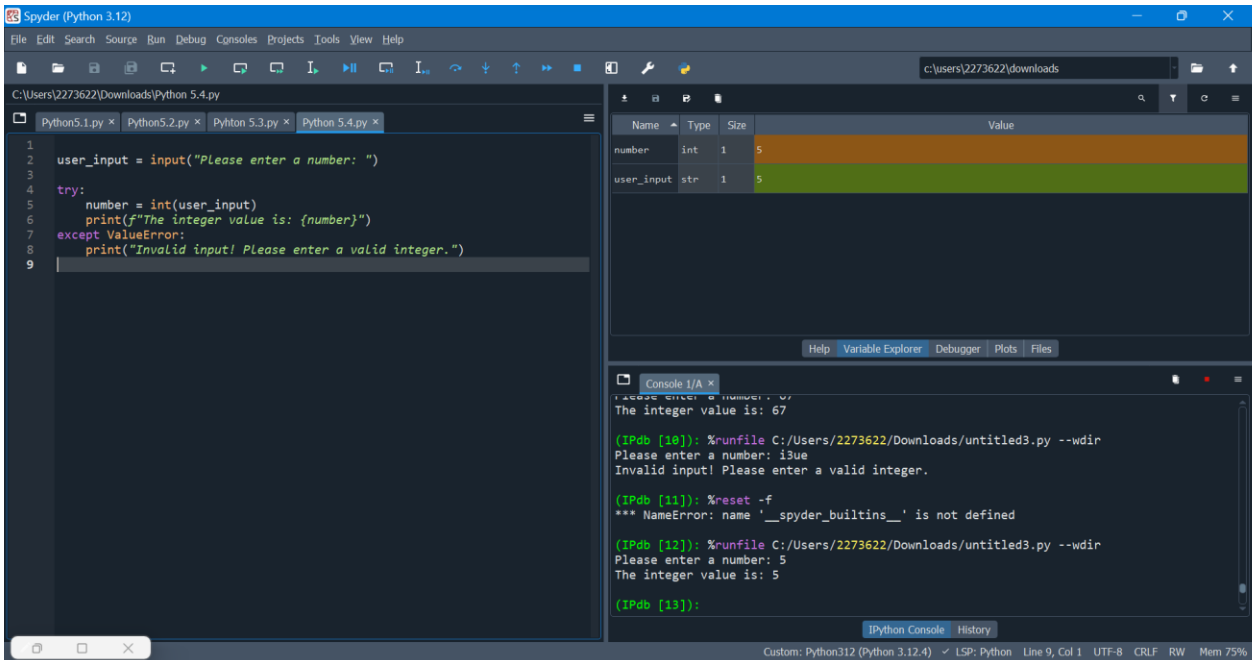
```
closed files")

_pickle.PicklingError: Cannot pickle closed files


(IPdb [8]): %runfile C:/Users/2273622/untitled2.py
--wdir
Total number of words: 230

(IPdb [9]):
```

Exercise 4:
Write a Python program that prompts the user to input a string and converts it to an integer. Use try-except blocks to handle any exceptions that might occur

Definition
This Python program prompts the user to input a string, attempts to convert the input to an integer, and handles any exceptions that might occur during the conversion process using try-except blocks.
Purpose
The primary purpose of this program is to:
1. **Get user input**: Prompt the user to enter a string.
2. **Convert input**: Attempt to convert the input string to an integer.
3. **Handle errors**: Gracefully handle any errors that occur during the conversion process, ensuring the program does not crash.

Use Cases
This type of program can be useful in various scenarios, such as:
1. **User Input Validation**: Ensuring that user inputs are valid integers before proceeding with further operations.
2. **Data Entry Applications**: Validating and processing numerical inputs in data entry forms.

Code:
```python
user_input = input("Please enter a number: ")
try:
    number = int(user_input)
    print(f"The integer value is: {number}")
except ValueError:
    print("Invalid input! Please enter a valid integer.")
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273622\downloads
C:\Users\2273622\Downloads\Python 5.4.py
Python5.1.py
Python5.2.py
Pyhton 5.3.py
Python 5.4.py
Help
Variable Explorer
Debugger
Plots
Files
Console 1/A
IPython Console
History
Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 9, Col 1 UTF-8 CRLF RW Mem 75%

```python
user_input = input("Please enter a number: ")

try:
    number = int(user_input)
    print(f"The integer value is: {number}")
except ValueError:
    print("Invalid input! Please enter a valid integer.")
```

Name	Type	Size	Value
number	int	1	5
user_input	str	1	5

```
The integer value is: 67

(IPdb [10]): %runfile C:/Users/2273622/Downloads/untitled3.py --wdir
Please enter a number: i3ue
Invalid input! Please enter a valid integer.

(IPdb [11]): %reset -f
*** NameError: name '__spyder_builtins__' is not defined

(IPdb [12]): %runfile C:/Users/2273622/Downloads/untitled3.py --wdir
Please enter a number: 5
The integer value is: 5

(IPdb [13]):
```

Exercise 5:

Write a Python program that prompts the user to input a list of integers and raises an exception if any of the integers in the list are negative.

Definition

This Python program prompts the user to input a list of integers, checks each integer in the list, and raises an exception if any of the integers are negative.

Purpose

The primary purpose of this program is to:

1. **Get user input**: Prompt the user to enter a list of integers.
2. **Validate input**: Check each integer in the list to ensure it is non-negative.
3. **Handle errors**: Raise an exception if any negative integers are found.

Use Cases

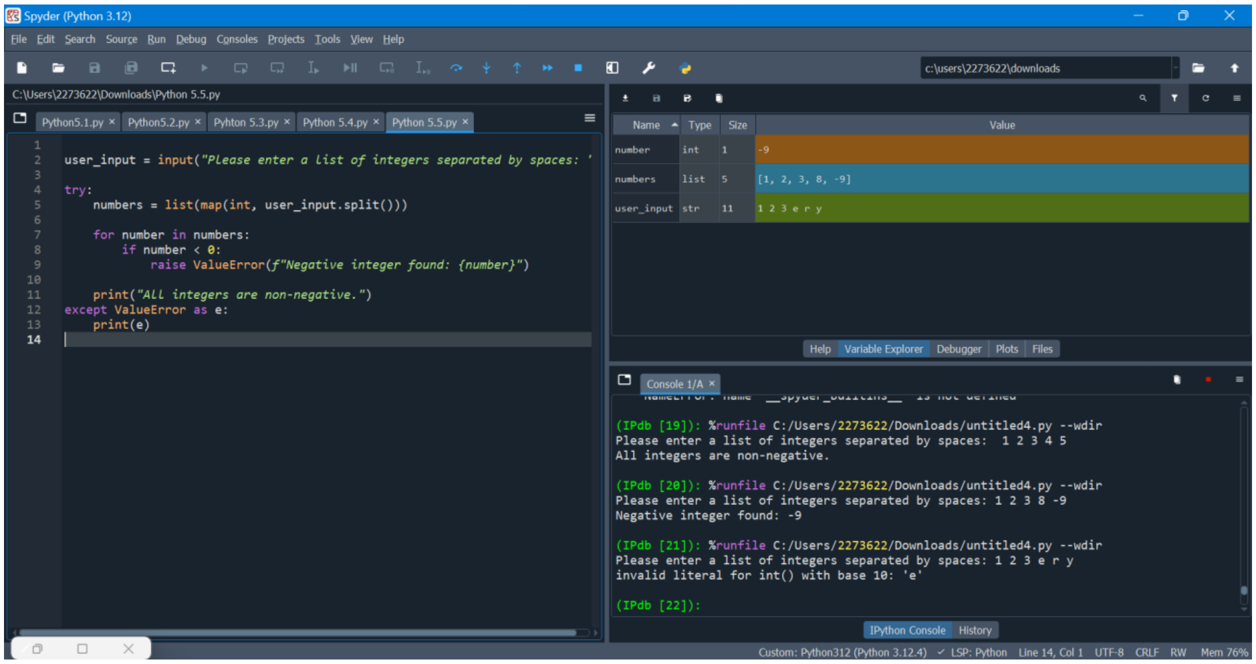
This type of program can be useful in various scenarios, such as:

1. **Data Validation**: Ensuring that a list of integers contains only non-negative values before processing.
2. **Input Sanitization**: Preventing negative values from being used in contexts where they are not allowed.

Code:

```
user_input = input("Please enter a list of integers separated by spaces: ")
try:
    numbers = list(map(int, user_input.split()))
    for number in numbers:
        if number < 0:
            raise ValueError(f"Negative integer found: {number}")
    print("All integers are non-negative.")
except ValueError as e:
```

    print(e)



Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273622\downloads
C:\Users\2273622\Downloads\Python 5.5.py
Python5.1.py Python5.2.py Pyhton 5.3.py Python 5.4.py Python 5.5.py
Help Variable Explorer Debugger Plots Files
Console 1/A
IPython Console History
Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 14, Col 1 UTF-8 CRLF RW Mem 76%

```python
user_input = input("Please enter a list of integers separated by spaces: '

try:
    numbers = list(map(int, user_input.split()))

    for number in numbers:
        if number < 0:
            raise ValueError(f"Negative integer found: {number}")

    print("All integers are non-negative.")
except ValueError as e:
    print(e)
```

Name	Type	Size	Value
number	int	1	-9
numbers	list	5	[1, 2, 3, 8, -9]
user_input	str	11	1 2 3 e r y

```
(IPdb [19]): %runfile C:/Users/2273622/Downloads/untitled4.py --wdir
Please enter a list of integers separated by spaces:  1 2 3 4 5
All integers are non-negative.

(IPdb [20]): %runfile C:/Users/2273622/Downloads/untitled4.py --wdir
Please enter a list of integers separated by spaces: 1 2 3 8 -9
Negative integer found: -9

(IPdb [21]): %runfile C:/Users/2273622/Downloads/untitled4.py --wdir
Please enter a list of integers separated by spaces: 1 2 3 e r y
invalid literal for int() with base 10: 'e'

(IPdb [22]):
```

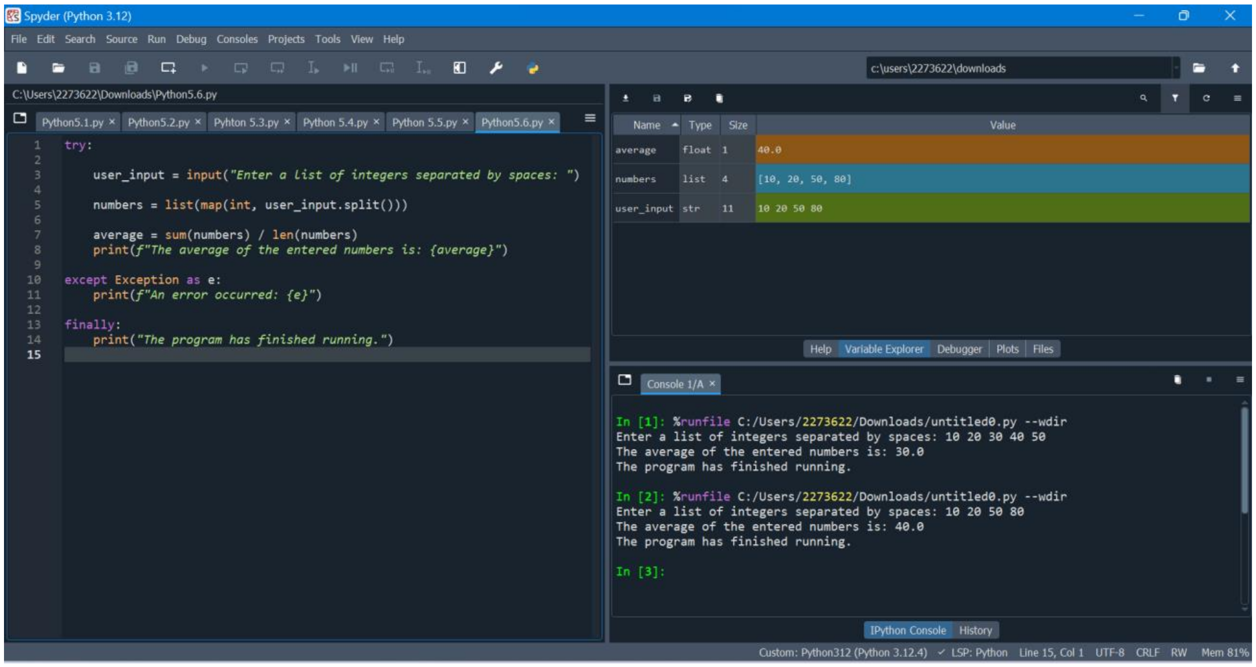
Exercise 6:
Write a Python program that prompts the user to input a list of integers and computes the average of those integers. Use try-except blocks to handle any exceptions that might occur.use the finally clause to print a message indicating that the program has finished running.

Definition
The Python program provided is a script that prompts the user to input a list of integers, computes the average of those integers, and handles potential exceptions that might occur during the process.
Purpose
The purpose of this program is to:
1. **Collect User Input**: Prompt the user to enter a list of integers.
2. **Process Data**: Convert the input string into a list of integers and compute the average.
3. **Handle Exceptions**: Use try-except blocks to manage errors such as invalid input or an empty list.
4. **Provide Feedback**: Inform the user of any errors and indicate when the program has finished running.

Use Cases
This program can be useful in various scenarios, such as:
1. **Data Processing**: Quickly calculating the average of a set of numbers provided by a user, which can be useful in data analysis tasks.
2. **User Interaction**: Demonstrating how to create interactive programs that respond to user input and handle errors gracefully.
3. **Debugging Practice**: Providing a simple example for practicing debugging and exception handling in Python.

Code:

```
try:
    user_input = input("Enter a list of integers separated by spaces: ")
    numbers = list(map(int, user_input.split()))
    average = sum(numbers) / len(numbers)
    print(f"The average of the entered numbers is: {average}")

except Exception as e:
    print(f"An error occurred: {e}")

finally:
    print("The program has finished running.")
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273622\downloads
C:\Users\2273622\Downloads\Python5.6.py
Python5.1.py Python5.2.py Pyhton 5.3.py Python 5.4.py Python 5.5.py Python5.6.py
Help Variable Explorer Debugger Plots Files
Console 1/A
IPython Console History

```
try:

    user_input = input("Enter a list of integers separated by spaces: ")

    numbers = list(map(int, user_input.split()))

    average = sum(numbers) / len(numbers)
    print(f"The average of the entered numbers is: {average}")

except Exception as e:
    print(f"An error occurred: {e}")

finally:
    print("The program has finished running.")
```

Name	Type	Size	Value
average	float	1	40.0
numbers	list	4	[10, 20, 50, 80]
user_input	str	11	10 20 50 80

```
In [1]: %runfile C:/Users/2273622/Downloads/untitled0.py --wdir
Enter a list of integers separated by spaces: 10 20 30 40 50
The average of the entered numbers is: 30.0
The program has finished running.

In [2]: %runfile C:/Users/2273622/Downloads/untitled0.py --wdir
Enter a list of integers separated by spaces: 10 20 50 80
The average of the entered numbers is: 40.0
The program has finished running.

In [3]:
```

Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 15, Col 1 UTF-8 CRLF RW Mem 81%

Exercise 7 :
Write a Python program that prompts the user to input a filename and writes a string to that file. Use try-except blocks to handle any exceptions that might occur and print a welcome message if there is no exception occurred.

Definition
The Python program provided is a script that prompts the user to input a filename and a string, writes the string to the specified file, and handles any exceptions that might occur during the process.
Purpose
The purpose of this program is to:
1. **Collect User Input**: Prompt the user to enter a filename and the content to be written to that file.

2. **File Operations**: Open the specified file in write mode and write the provided content to it.
3. **Handle Exceptions**: Use try-except blocks to manage errors such as file access issues.
4. **Provide Feedback**: Inform the user of any errors and print a welcome message if the operation is successful.

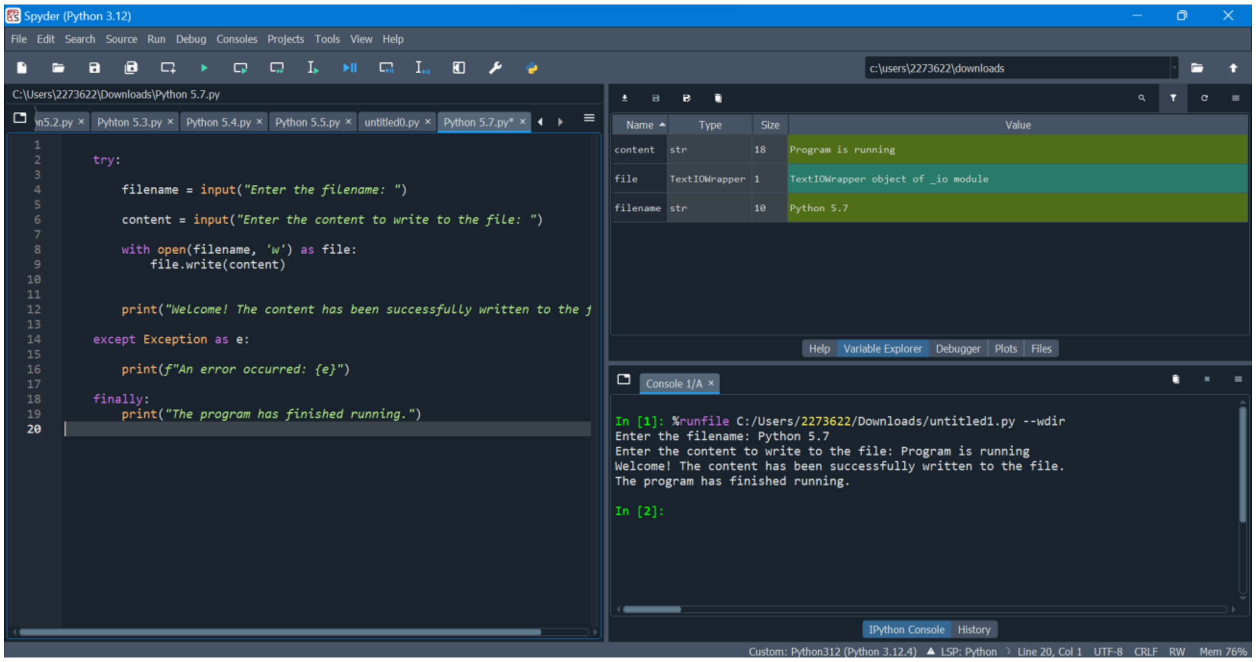
Use Cases

This program can be useful in various scenarios, such as:

1. **Data Storage**: Saving user-generated content to a file for later use or processing.
2. **Configuration Files**: Creating or updating configuration files based on user input.
3. **Logging**: Writing logs or notes to a file, which can be useful for debugging or record-keeping.

Code:

```
try:
    filename = input("Enter the filename: ")
    content = input("Enter the content to write to the file: ")
    with open(filename, 'w') as file:
        file.write(content)

    print("Welcome! The content has been successfully written to the file.")

  except Exception as e:
    print(f"An error occurred: {e}")
  finally:
    print("The program has finished running.")
```



C:\Users\2273622\Downloads\Python 5.7.py

```
try:

    filename = input("Enter the filename: ")

    content = input("Enter the content to write to the file: ")

    with open(filename, 'w') as file:
        file.write(content)


    print("Welcome! The content has been successfully written to the f

except Exception as e:

    print(f"An error occurred: {e}")

finally:
    print("The program has finished running.")
```

Name	Type	Size	Value
content	str	18	Program is running
file	TextIOWrapper	1	TextIOWrapper object of _io module
filename	str	10	Python 5.7

```
In [1]: %runfile C:/Users/2273622/Downloads/untitled1.py --wdir
Enter the filename: Python 5.7
Enter the content to write to the file: Program is running
Welcome! The content has been successfully written to the file.
The program has finished running.

In [2]:
```