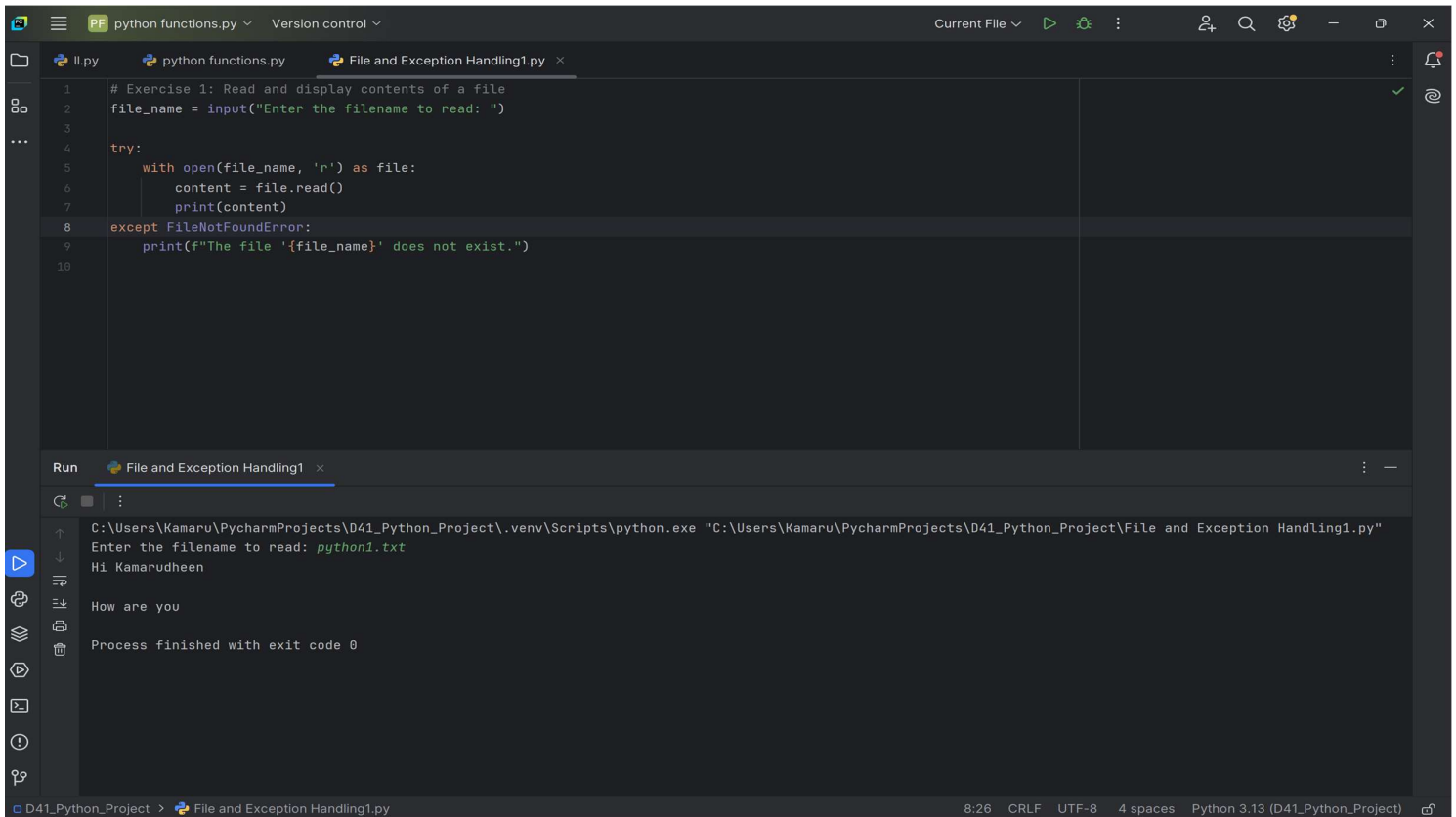


EXERCISE 1



The screenshot shows the PyCharm IDE with a file named `File and Exception Handling1.py`. The code implements a function to read a file and handle a `FileNotFoundError`. The execution output shows the program running successfully, reading the contents of `python1.txt`.

```
1 # Exercise 1: Read and display contents of a file
2 file_name = input("Enter the filename to read: ")
3
4 try:
5     with open(file_name, 'r') as file:
6         content = file.read()
7         print(content)
8 except FileNotFoundError:
9     print(f"The file '{file_name}' does not exist.")
10
```

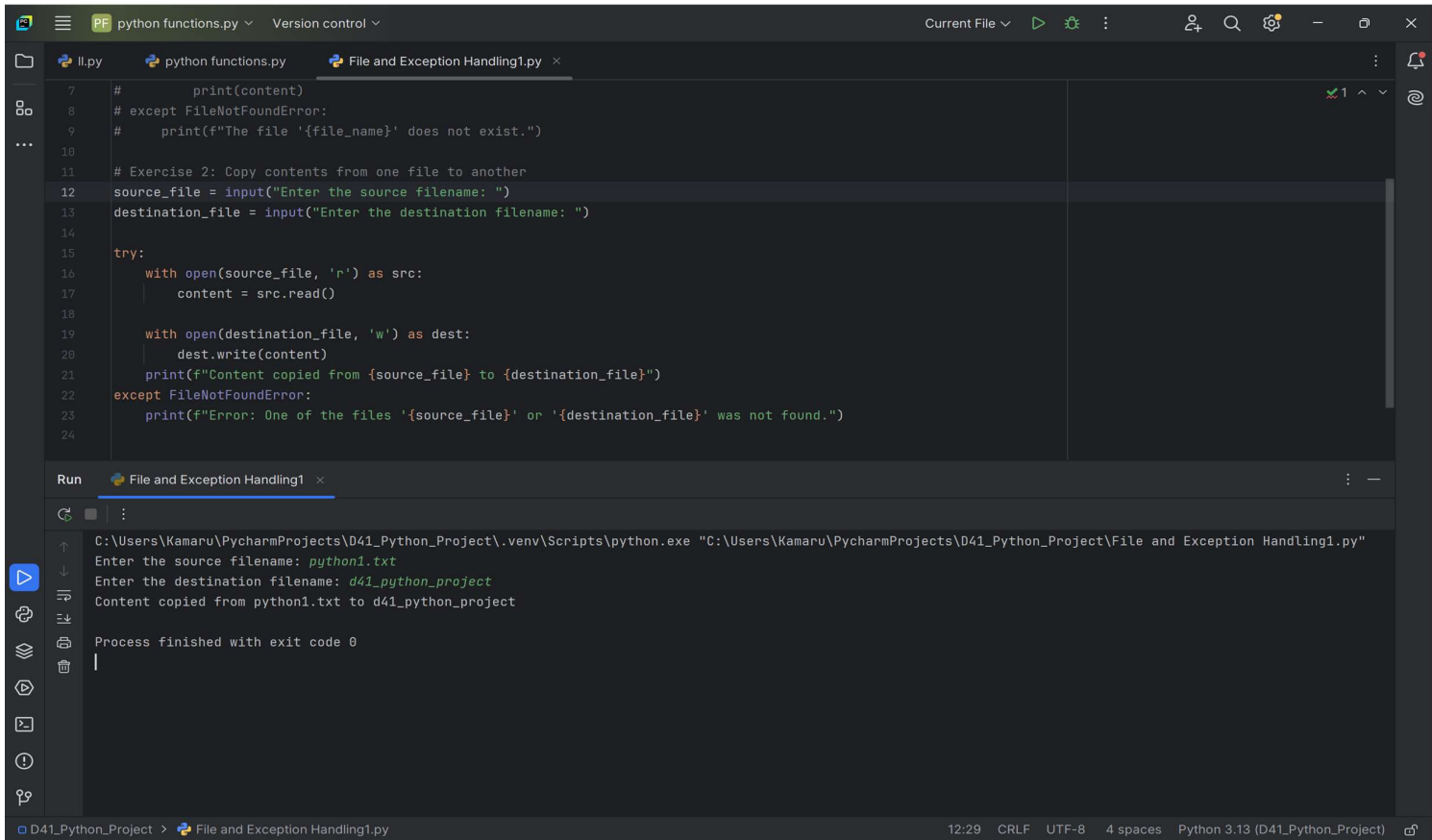
Run File and Exception Handling1

```
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"
Enter the filename to read: python1.txt
Hi Kamarudheen

How are you

Process finished with exit code 0
```

EXERCISE 2



The screenshot shows the PyCharm IDE with a file named `File and Exception Handling1.py`. The code implements a function to copy the contents of one file to another, handling a `FileNotFoundError`. The execution output shows the program running successfully, copying the contents of `python1.txt` to `d41_python_project`.

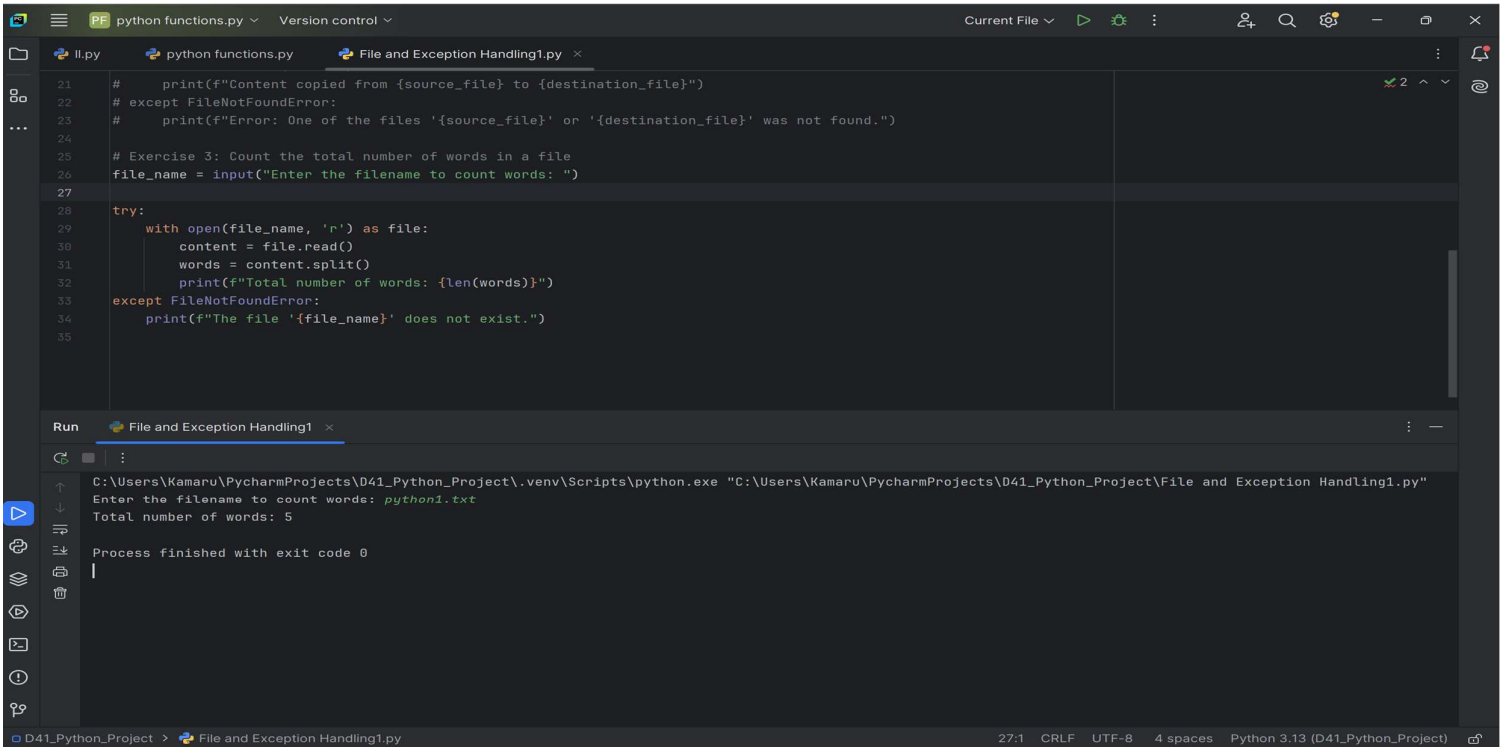
```
7 # print(content)
8 # except FileNotFoundError:
9 #     print(f"The file '{file_name}' does not exist.")
10
11 # Exercise 2: Copy contents from one file to another
12 source_file = input("Enter the source filename: ")
13 destination_file = input("Enter the destination filename: ")
14
15 try:
16     with open(source_file, 'r') as src:
17         content = src.read()
18
19     with open(destination_file, 'w') as dest:
20         dest.write(content)
21     print(f"Content copied from {source_file} to {destination_file}")
22 except FileNotFoundError:
23     print(f"Error: One of the files '{source_file}' or '{destination_file}' was not found.")
24
```

Run File and Exception Handling1

```
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"
Enter the source filename: python1.txt
Enter the destination filename: d41_python_project
Content copied from python1.txt to d41_python_project

Process finished with exit code 0
```

EXERCISE 3



The screenshot shows the PyCharm IDE with a Python file named 'File and Exception Handling1.py'. The code defines a function to count words in a file, using a try-except block to handle FileNotFound errors. The Run console shows the program execution with the input 'python1.txt' and the output 'Total number of words: 5'.

```
21 # print(f"Content copied from {source_file} to {destination_file}")
22 # except FileNotFoundError:
23 #     print(f"Error: One of the files '{source_file}' or '{destination_file}' was not found.")
24
25 # Exercise 3: Count the total number of words in a file
26 file_name = input("Enter the filename to count words: ")
27
28 try:
29     with open(file_name, 'r') as file:
30         content = file.read()
31         words = content.split()
32         print(f"Total number of words: {len(words)}")
33 except FileNotFoundError:
34     print(f"The file '{file_name}' does not exist.")
35
```

Run File and Exception Handling1

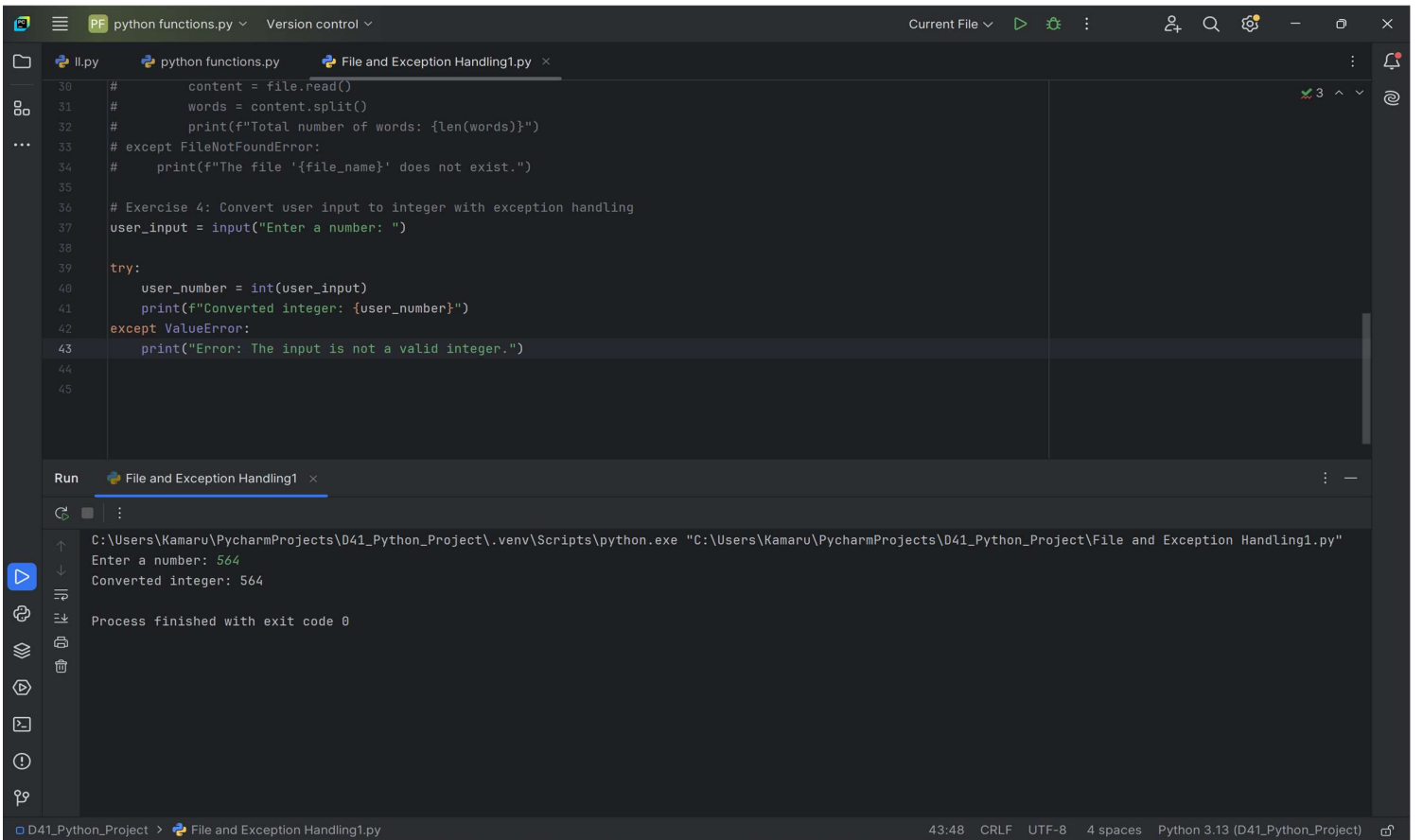
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"

Enter the filename to count words: *python1.txt*

Total number of words: 5

Process finished with exit code 0

EXERCISE 4



The screenshot shows the PyCharm IDE with a Python file named 'File and Exception Handling1.py'. The code defines a function to convert user input to an integer, using a try-except block to handle ValueError exceptions. The Run console shows the program execution with the input '564' and the output 'Converted integer: 564'.

```
30 # content = file.read()
31 # words = content.split()
32 # print(f"Total number of words: {len(words)}")
33 # except FileNotFoundError:
34 #     print(f"The file '{file_name}' does not exist.")
35
36 # Exercise 4: Convert user input to integer with exception handling
37 user_input = input("Enter a number: ")
38
39 try:
40     user_number = int(user_input)
41     print(f"Converted integer: {user_number}")
42 except ValueError:
43     print("Error: The input is not a valid integer.")
44
45
```

Run File and Exception Handling1

C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"

Enter a number: *564*

Converted integer: 564

Process finished with exit code 0

```
30 # content = file.read()
31 # words = content.split()
32 # print(f"Total number of words: {len(words)}")
33 # except FileNotFoundError:
34 #     print(f"The file '{file_name}' does not exist.")
35
36 # Exercise 4: Convert user input to integer with exception handling
37 user_input = input("Enter a number: ")
38
39 try:
40     user_number = int(user_input)
41     print(f"Converted integer: {user_number}")
42 except ValueError:
43     print("Error: The input is not a valid integer.")
44
45
```

Run File and Exception Handling1

C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"

Enter a number: *karam*

Error: The input is not a valid integer.

Process finished with exit code 0

D41_Python_Project > File and Exception Handling1.py 43:48 CRLF UTF-8 4 spaces Python 3.13 (D41_Python_Project)

EXERCISE 5

```
45 # Exercise 5: Raise exception for negative numbers in the list
46 try:
47     user_input = input("Enter a list of integers (comma-separated): ")
48     numbers = list(map(int, user_input.split(',')))
49
50     for number in numbers:
51         if number < 0:
52             raise ValueError("Negative number found.")
53
54     print("All numbers are non-negative.")
55 except ValueError as e:
56     print(f"Error: {e}")
57
58
```

Run File and Exception Handling1

C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"

Enter a list of integers (comma-separated): *6,8,-2*

Error: Negative number found.

Process finished with exit code 0

D41_Python_Project > File and Exception Handling1.py 51:23 CRLF UTF-8 4 spaces Python 3.13 (D41_Python_Project)

The screenshot shows the PyCharm IDE with a file named `File and Exception Handling1.py`. The code implements Exercise 5, which prompts the user for a list of integers and checks for negative values. The code is as follows:

```
45 # Exercise 5: Raise exception for negative numbers in the list
46 try:
47     user_input = input("Enter a list of integers (comma-separated): ")
48     numbers = list(map(int, user_input.split(',')))
49
50     for number in numbers:
51         if number < 0:
52             raise ValueError("Negative number found.")
53
54     print("All numbers are non-negative.")
55 except ValueError as e:
56     print(f"Error: {e}")
57
58
```

The Run window shows the following output:

```
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"
Enter a list of integers (comma-separated): 6,3,4
All numbers are non-negative.
Process finished with exit code 0
```

EXERCISE 6

The screenshot shows the PyCharm IDE with the same file `File and Exception Handling1.py`. The code implements Exercise 6, which computes the average of a list of integers and includes a finally block to ensure the program prints a completion message. The code is as follows:

```
55 # except ValueError as e:
56 #     print(f"Error: {e}")
57
58 # Exercise 6: Compute average with exception handling and finally block
59 try:
60     user_input = input("Enter a list of integers (comma-separated): ")
61     numbers = list(map(int, user_input.split(',')))
62
63     if len(numbers) == 0:
64         raise ValueError("The list cannot be empty.")
65
66     avg = sum(numbers) / len(numbers)
67     print(f"Average of the numbers: {avg}")
68 except ValueError as e:
69     print(f"Error: {e}")
70 finally:
71     print("Program execution is complete.")
72
```

The Run window shows the following output:

```
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"
Enter a list of integers (comma-separated): 4,8,7,6
Average of the numbers: 6.25
Program execution is complete.
Process finished with exit code 0
```

EXERCISE 7

The screenshot displays the PyCharm IDE interface. The top toolbar includes icons for file operations, running, and debugging. The main editor window shows a Python file named 'File and Exception Handling1.py' with the following code:

```
66 # avg = sum(numbers) / len(numbers)
67 # print(f"Average of the numbers: {avg}")
68 # except ValueError as e:
69 #     print(f"Error: {e}")
70 # finally:
71 #     print("Program execution is complete.")
72
73 # Exercise 7: Write a string to a file with exception handling
74 file_name = input("Enter the filename to write to: ")
75 content = input("Enter the string to write into the file: ")
76
77 try:
78     with open(file_name, 'w') as file:
79         file.write(content)
80     print("Welcome! Content successfully written to the file.")
81 except Exception as e:
82     print(f"An error occurred: {e}")
83
84
```

Below the editor is the 'Run' console, which shows the execution of the script. The output indicates that the file 'python1.txt' was successfully created and the content 'i am fine' was written to it. The process finished with an exit code of 0.

```
Run File and Exception Handling1
C:\Users\Kamaru\PycharmProjects\D41_Python_Project\.venv\Scripts\python.exe "C:\Users\Kamaru\PycharmProjects\D41_Python_Project\File and Exception Handling1.py"
Enter the filename to write to: python1.txt
Enter the string to write into the file: i am fine
Welcome! Content successfully written to the file.

Process finished with exit code 0
```

The bottom status bar shows the project path 'D41_Python_Project', the file name 'File and Exception Handling1.py', and various settings like '76:1 CRLF UTF-8 4 spaces Python 3.13 (D41_Python_Project)'.