# Week-5 and Week-6

1. File name: count\_text.py

**Description:**

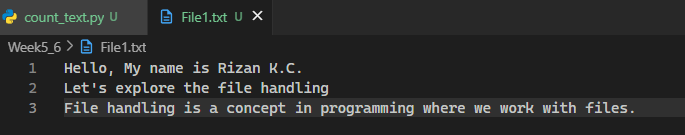
This program counts the number of lines, words and character present in the text file.

This program uses file methods to accomplish the task.

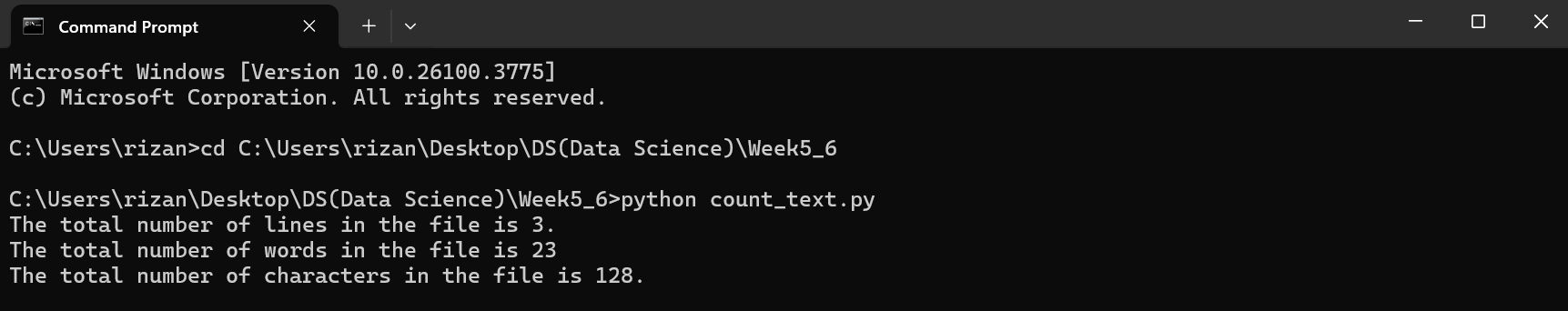
Test is carried out.

**Test:**

Contents present in the file:



Running the program:



Based on the test and the output, the program is working properly.

1. File name: copy\_content.py

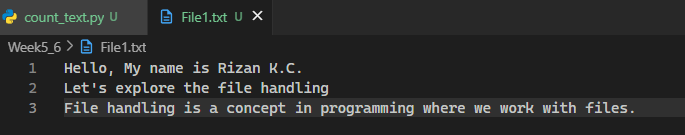
**Description:**

The program copies the content of one file and paste it to another file with the use of file handling concepts.

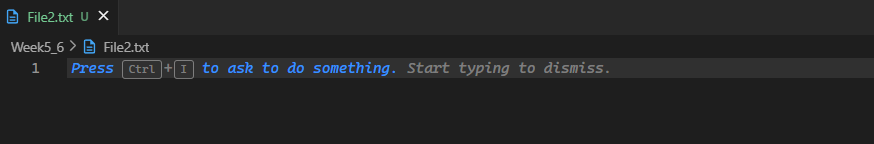
The program works on two files named File1.txt and File2.txt. File1.txt is the file where the content is being copied from while File2.txt is an empty where the content copied is pasted.

**Test:**

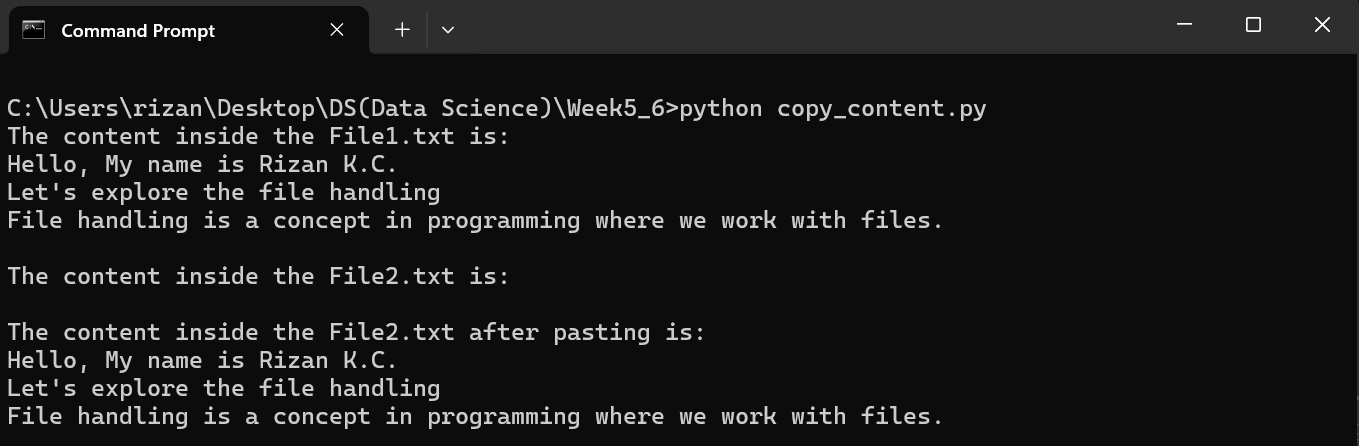
Content of File1.txt:



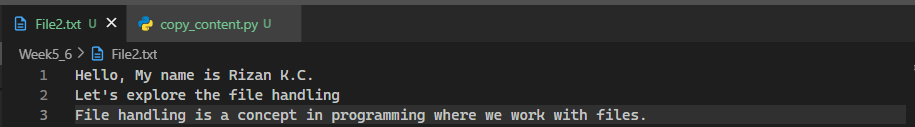
Content of File2.txt(empty):



Running the program:



Contents of File2.txt after pasting:



Based on the test, the program works properly.

1. File name: file\_replaceword.py

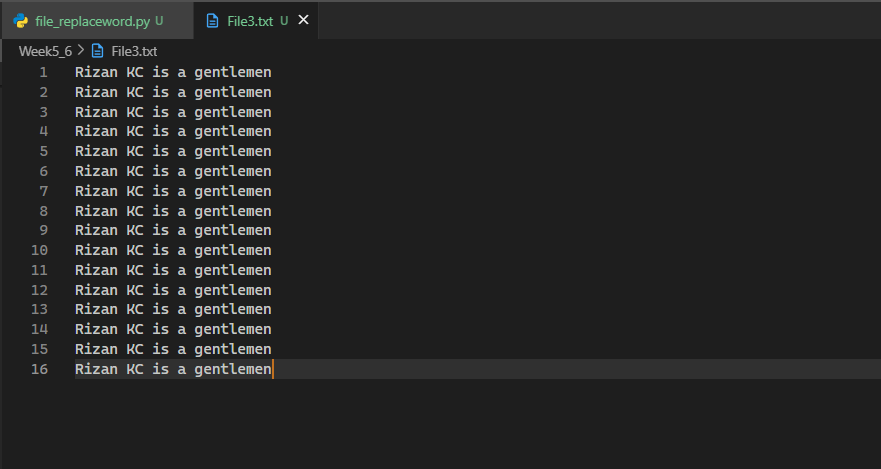
**Description:**

This program finds the specific word in the file and replace that word with another word in the file.

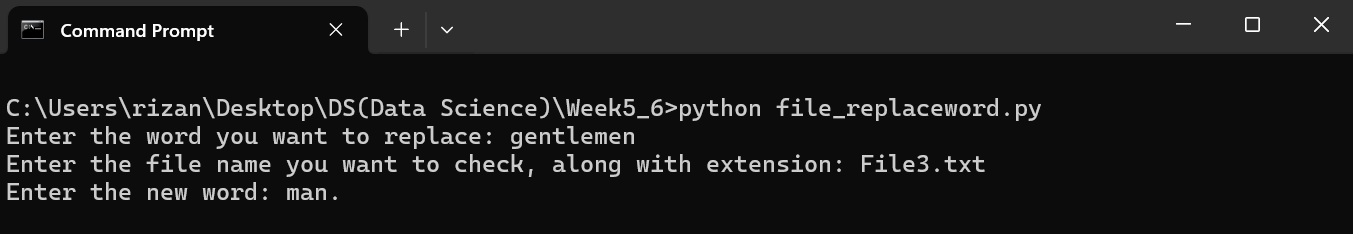
Program prompts the user for two inputs: word to replace and file name. Then, it checks if the word exists in the file or not. If exists, it asks for the replacement word otherwise it again prompts for a different word.

**Test**:

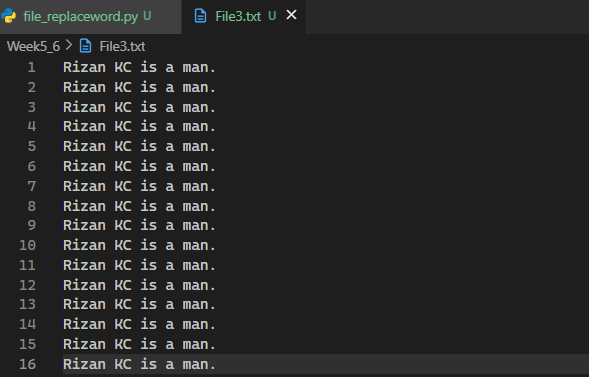
Content inside the File3.txt:



Running the program:



Content inside the File3.txt:



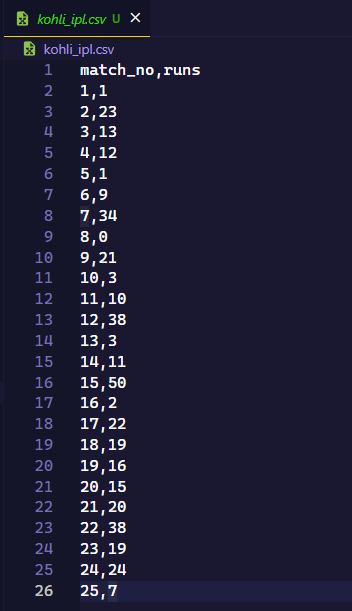
Based on the test, the program works properly.

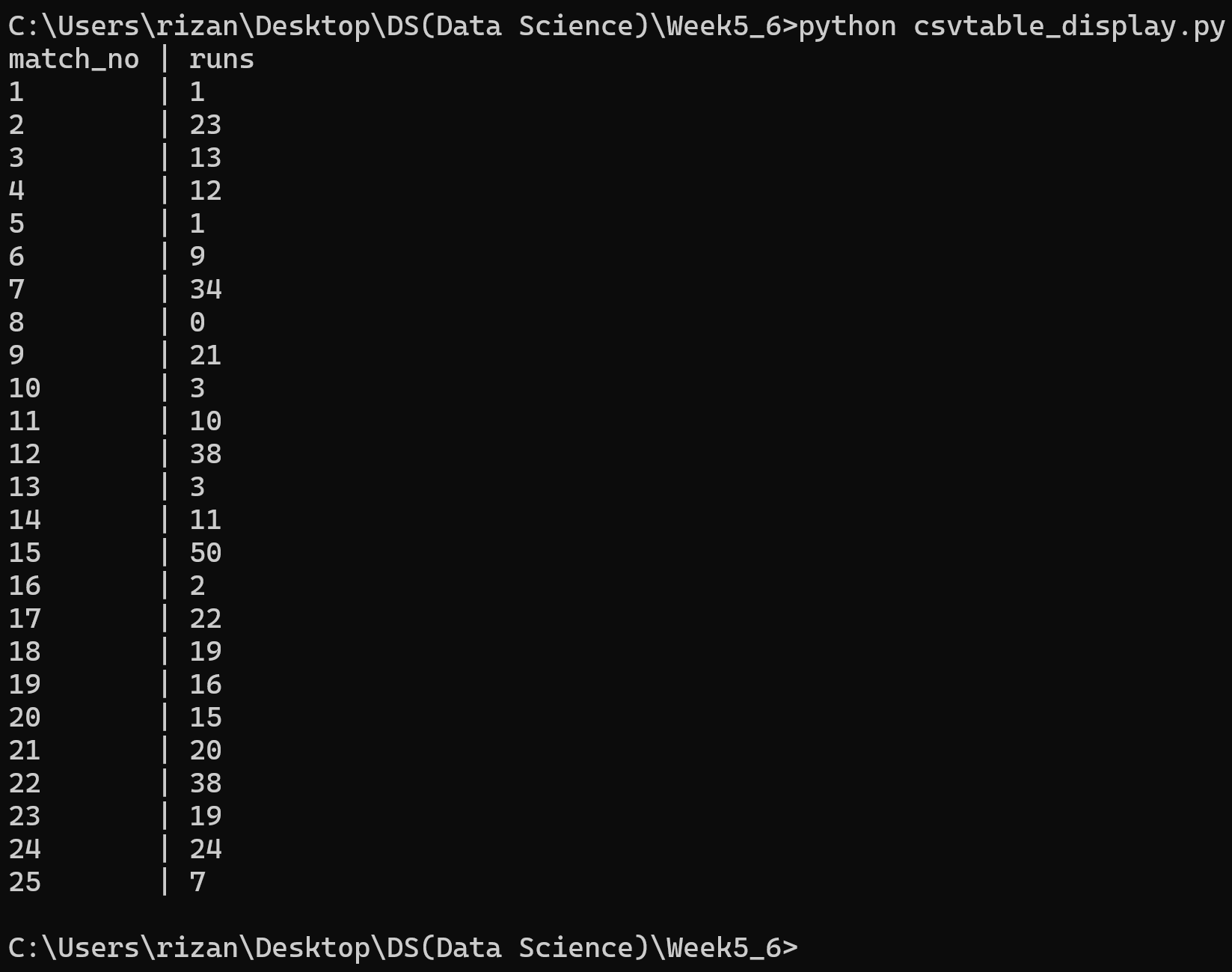
1. File name: csvtable\_display.py

**Description:**

This program opens the csv file which contains 2 columns, one being match\_no while other is runs made in the corresponding match\_no. It displays the file in a tabular form.

**Test:**

****

****

Based on the test, the program works properly.

1. File name: count\_occurence.py

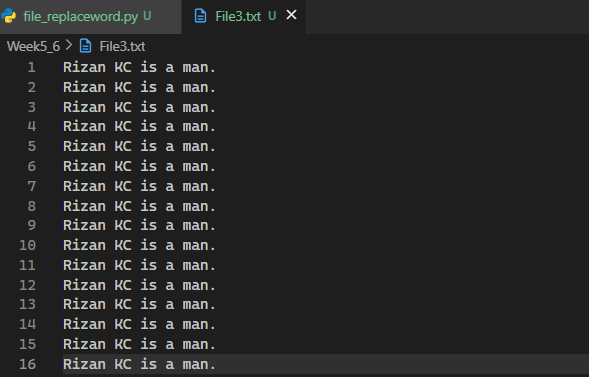
**Description:**

The program counts the occurrence of each word in a file. TO solve the problem, we have used a dictionary where each words are stored as key and the count is stored as value.

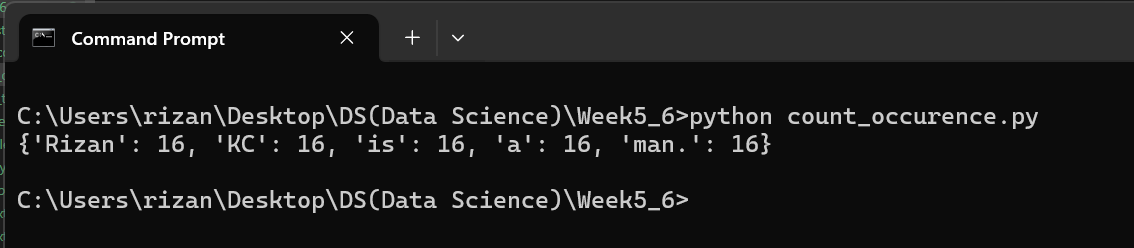
Test is carried out.

**Test:**

Content inside the File3.txt



Running the program should show each word as key and the value should sum up to 16 as each word counts to 16.

****

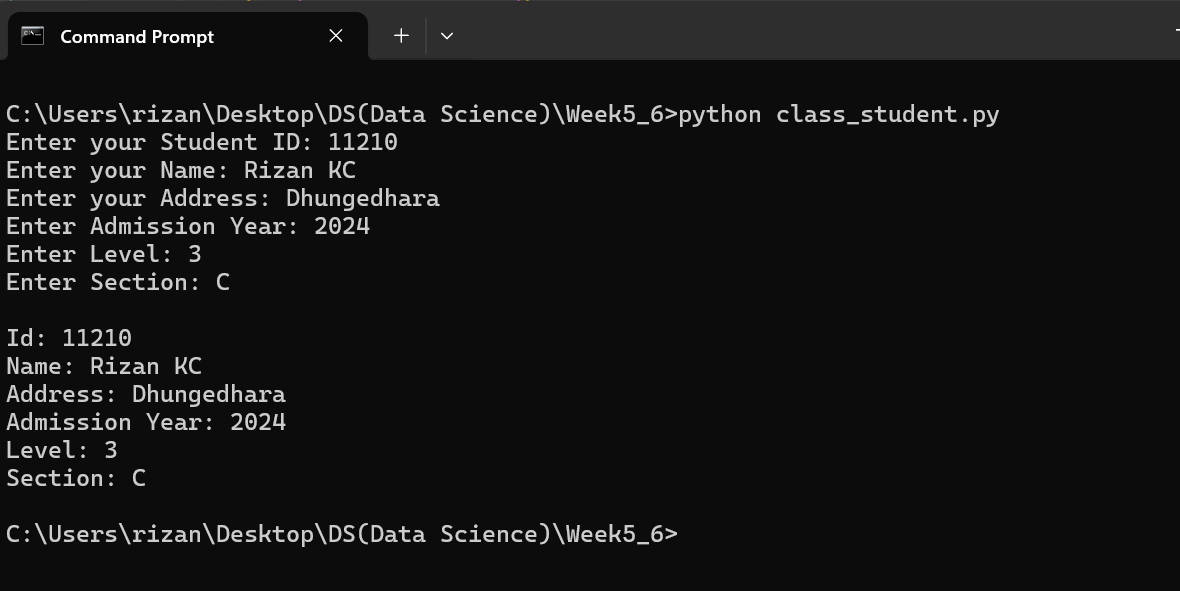
Based on the test, the program works properly.

1. File name: class\_student.py

**Description:**

The program creates a class Student with the attributes such as id, name, address, admission\_year, level, section. This creates the object by taking attribute value from the use and displays each of the attribute provide in the input.

**Test:**

****

Based on the test, the program works properly.

1. File name: class\_Employment.py

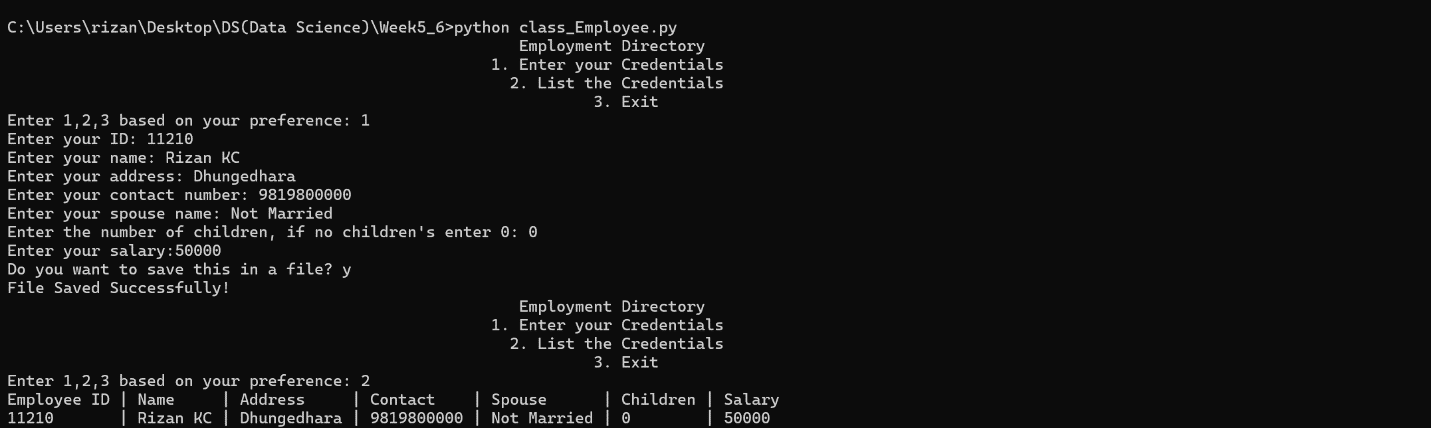
**Description:**

This program takes the input for multiple employees, saves them in a csv file named “employees.csv” and displays the content of the file.

This program creates a Class named Employee where different attribute values are taken as input and saved in the csv file. Later, the files content also can be viewed from the program.

**Test:**

**Initial Test(No file named “employees.csv” exists):**

**A computer screen with white text

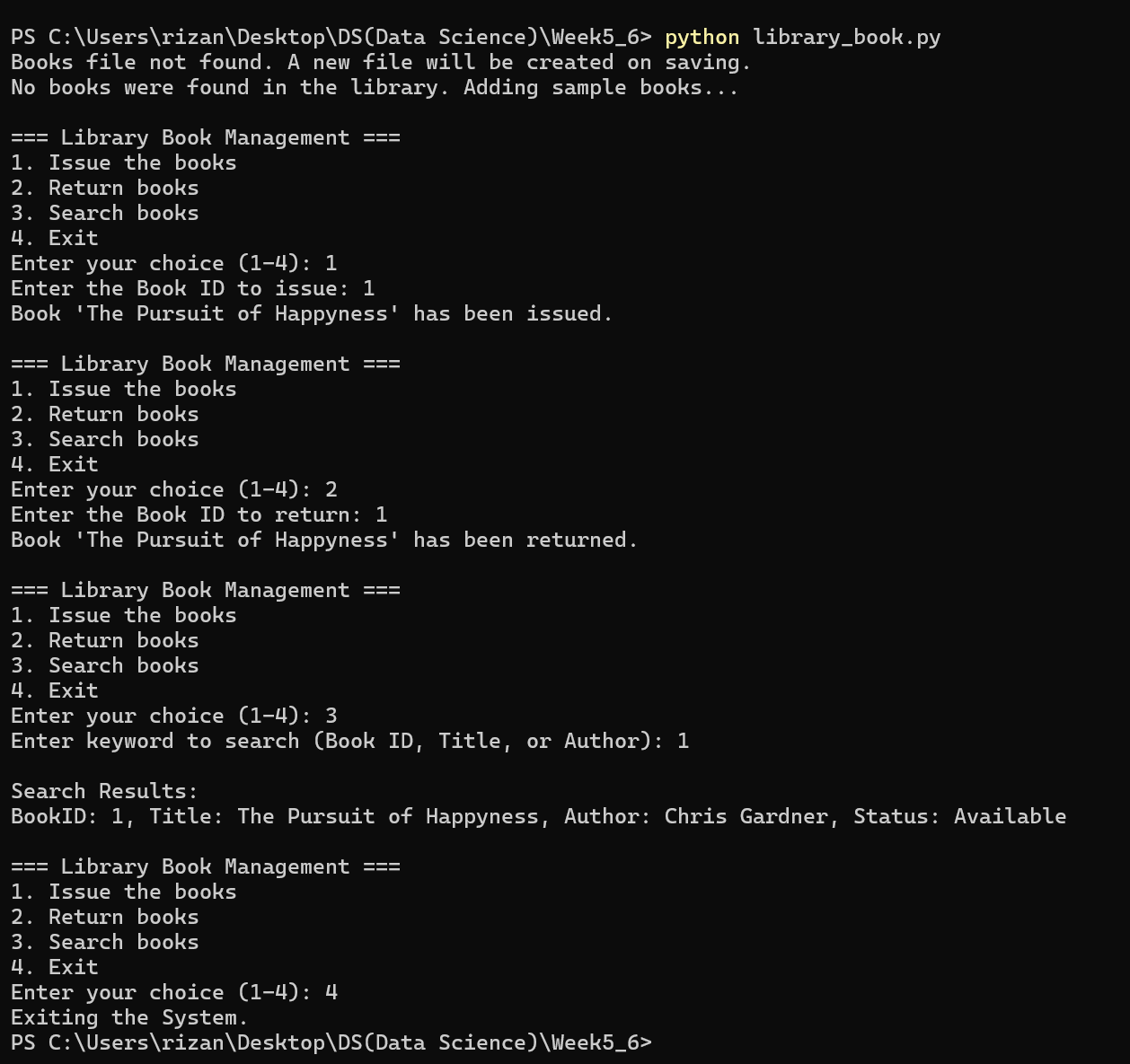
AI-generated content may be incorrect.**

1. File name: library\_book.py

**Description:**

The program has a class named Book which has attributes like id, name, author, and is the book issued bool attribute (default ‘False’). Along with this, the class has methods like issue the books, return the issued book, search books using id, title and author of the book.

The program user OOP and file handling concepts. While issuing the book, the file is updated where the is\_issued value is changed to True and viceversa, returning the book.



Based on the program, the program works properly.