



slington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CS4051NI Fundamentals of Computing

Assessment Weightage & Type

60% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Sujen Shrestha

Group: N4

London Met ID: 20049250

College ID: NP01NT4S210105

Assignment Due Date: September 10, 2021

Assignment Submission Date: September 10, 2021

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

1. Introduction	1
1.1 Introduction to the project.....	1
1.2 Goals and Objectives	1
2. Discussion and Analysis	2
2.1 Algorithm.....	2
2.2 Flowchart.....	4
2.3 Pseudocode	8
2.3.1 main.py.....	8
2.3.2 book_borrow.py.....	9
2.3.3 book_return.py	13
2.3.4 functions.py	16
2.3.5 messages.py	17
2.4 Data Structures	18
2.4.1 Integer	18
2.4.2 String	18
2.4.3 Boolean	19
2.4.4 Dictionary.....	19
2.4.5 List.....	20
3. Program	21
4. Testing	52
5. Conclusion	74
6. Appendix.....	75
main.py	75
book_borrow.py	76

book_return.py.....	80
functions.py	84
messages.py	84
Bibliography	88

List of Figures

Figure 1: Flowchart of program	4
Figure 2: Flowchart of program from connector A	5
Figure 3: Flowchart of program from connector B	6
Figure 4: Flowchart of program from connector C	7
Figure 5: Integer data structure in python.....	18
Figure 6: String data structure in python	19
Figure 7: Boolean data structure in python.....	19
Figure 8: Dictionary data structure in python.....	20
Figure 9: List data structure in python	20
Figure 10: User Interface after starting the program	21
Figure 11: Stock file containing book details	22
Figure 12: Output after receiving string input as value	23
Figure 13: Output after receiving unspecified number as value	24
Figure 14: Output after receiving 1 as value.....	25
Figure 15: Output after receiving book ID as input where quantity is 0	26
Figure 16: Output after receiving string input as book ID	27
Figure 17: Output after receiving unspecified number as book ID.....	28
Figure 18: Output after receiving 0 as book ID.....	29
Figure 19: Output after receiving appropriate book ID as input	30
Figure 20: Output after receiving borrower's name	31
Figure 21: Stock file after a book is borrowed	32
Figure 22: Output after receiving 'y' as input to borrow another book	33
Figure 23: Output after borrowing another book.....	34
Figure 24: Stock file after another book is borrowed	35
Figure 25: Output after borrow completion	36
Figure 26: Bill generated after borrowing the books	37
Figure 27: Output after receiving 2 as input in value	38
Figure 28: Output after receiving unknown customer name	39
Figure 29: Output after receiving valid customer name	40
Figure 30: Bill generated after returning the books	41

Figure 31: Output after receiving valid customer name continued	42
Figure 32: Stock file after returning the books	43
Figure 33: Output when multiple customer name is same.....	44
Figure 34: Output when string input is received as customer ID	45
Figure 35: Output when unspecified number is received as customer ID.....	46
Figure 36: Output when appropriate customer ID is received	47
Figure 37: Bill generated after returning the books late	48
Figure 38: Output when appropriate customer ID is received continued	49
Figure 39: Stock file after all books are returned.....	50
Figure 40: Output when 3 is received as value	51
Figure 41: Test – Program running as required.....	53
Figure 42: Test – Error displayed when sting input	54
Figure 43: Test – Error displayed when unspecified number input as value	55
Figure 44: Test – Error message displayed when string input as book ID.....	56
Figure 45: Test – Error displayed when unspecified number input as book ID.....	57
Figure 46: Test – Error displayed when incorrect borrower name entered.....	58
Figure 47: Test - Output after receiving 1 as value	59
Figure 48:Test - Output after receiving borrower's name	60
Figure 49: Test - Stock file after a book is borrowed	60
Figure 50: Test - Output after receiving 'y' as input to borrow another book	61
Figure 51: Test - Output after borrowing another book.....	62
Figure 52: Test - Stock file after another book is borrowed	62
Figure 53: Test - Output after borrow completion	63
Figure 54: Test - Bill generated after borrowing the books.....	64
Figure 55: Test - Output after receiving 2 as input in value	65
Figure 56: Test - Output after receiving unknown customer name	66
Figure 57: Test - Output after receiving valid customer name	67
Figure 58: Test - Bill generated after returning the books	68
Figure 59: Test - Output after receiving valid customer name continued	69
Figure 60: Test - Stock file after returning the books.....	70
Figure 33: Test - Output when multiple customer name is same	71

Figure 36: Test - Output when appropriate customer ID is received	72
Figure 37: Test - Bill generated after returning the books late.....	73

List of Tables

Table 1: To test the implementation of try and except.....	52
Table 2: To test the result when incorrect id or name is entered.....	56
Table 3: To test the complete process of borrowing the books	59
Table 4: To test the complete process of returning the books.....	65
Table 5: To test the return process after lending period is over	71

1. Introduction

1.1 Introduction to the project

The project is about creating a library management system which helps to keep the records of basic library transactions like borrowing and returning of books. The project was created by using various tools like python, IDLE, draw.io and MS Word. In the program, the stock of books is displayed to the user so that they can borrow their required books if it is available. The number of books borrowed by a customer gets reduced from the stock and a bill is generated for the customer which contains all the details of the transaction (borrowing). Likewise, when a customer returns the books, another bill is generated for the customer which contains all the details of the transaction (return). The lending period of books is 10 days. If the customer returns the books after the lending period is over (i.e., 10 days), they have to pay a fine of \$0.25 for each passing day after the lending period. This information is also recorded in the bill (text file). The stock is increased for the books which have been returned. A text file with unique filename is generated for all the transactions carried out through the program.

1.2 Goals and Objectives

- To learn about computing and programming language.
- To learn about the various built-in keywords in python.
- To learn about different data structures in python.
- To learn about the conditional statements and use them as required.
- To learn about control flow statements and use them to iterate a process until the required condition is achieved.
- To develop the logic and create algorithm, flowchart and pseudocode for the program.
- To learn about functions and implement modularity in the program.
- To understand about errors and exceptions and handle them effectively.
- To develop a library management system.
- To test the program and see if it is functioning as required.

2. Discussion and Analysis

2.1 Algorithm

An algorithm is a structured process of solving a recurrent problem. It is designed in such a way that it conducts a sequence of specified actions. (TechTarget Contributor, 2021)

The algorithm used for developing this program is described below:

Step 1: Start

Step 2: Display Welcome Message

Step 3: Read books.txt file

Step 4: continueLoop = True

Step 5: WHILE continueLoop == True

Step 6: Display output from books.txt file

Step 7: Print "Enter '1' to borrow"

Step 8: Print "Enter '2' to return"

Step 9: Print "Enter '3' to exit"

Step 10: Input value

Step 11: IF value == 1,

 Step 12: borrowLoop = True

 Step 13: WHILE borrowLoop == True

 Step 14: Print "Enter Book ID"

 Step 15: Input Book ID

 Step 16: IF b == 0,

 Step 16.1: Return to Step 6

 Step 17: ELIF b == key,

 Step 17.1: Display borrow information

 Step 17.2: Print "Do you want to borrow another book?"

 Step 17.3: Input answer

 Step 17.4: IF answer != "y",

 Step 17.4.1: Write borrow text file

Step 17.4.2: Update books.txt
Step 17.4.3: borrowLoop = False
Step 17.4.4: Display borrow details
Step 17.4.5: Return to Step 6
Step 17.5: ELSE, Return to Step 14
Step 18: ELSE, Print "Provide valid ID"; Return to Step 14
Step 19: ELIF value == 2,
Step 20: returnLoop = True
Step 21: WHILE returnLoop == True
Step 22: Print "Enter name of customer"
Step 23: Input name
Step 24: IF name == customer,
Step 24.1: IF len(customer) > 1,
Step 24.1.1: Display files matching customer name
Step 24.1.2: Input Customer ID
Step 24.1.3: Write return text file
Step 24.1.4: Update books.txt
Step 24.1.5: returnLoop = False
Step 24.1.6: Display return details
Step 24.1.7: Return to Step 6
Step 24.2: ELSE, Go to Step 24.1.3
Step 25: ELIF value == 3,
Step 26: Display exit message
Step 27: continueLoop = False
Step 28: Stop
Step 29: ELSE; Print "Invalid input"; Return to Step 6

2.2 Flowchart

A flowchart is a representation of a process in graphical or symbolic form. Each step in the process is represented by a specific symbol and includes a brief description of the step. The flowchart elements are connected by arrows which indicate the direction of the process flow. (Hebb, 2021)

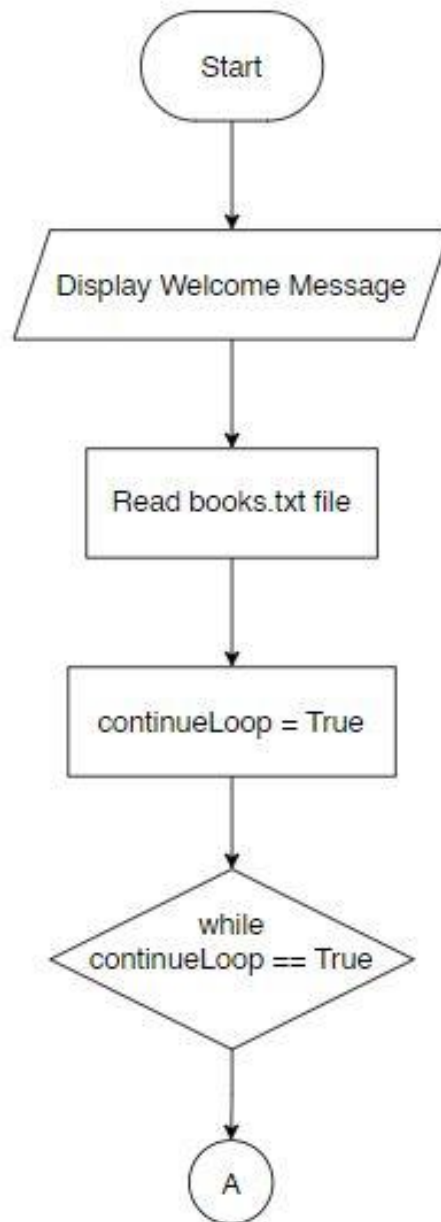


Figure 1: Flowchart of program

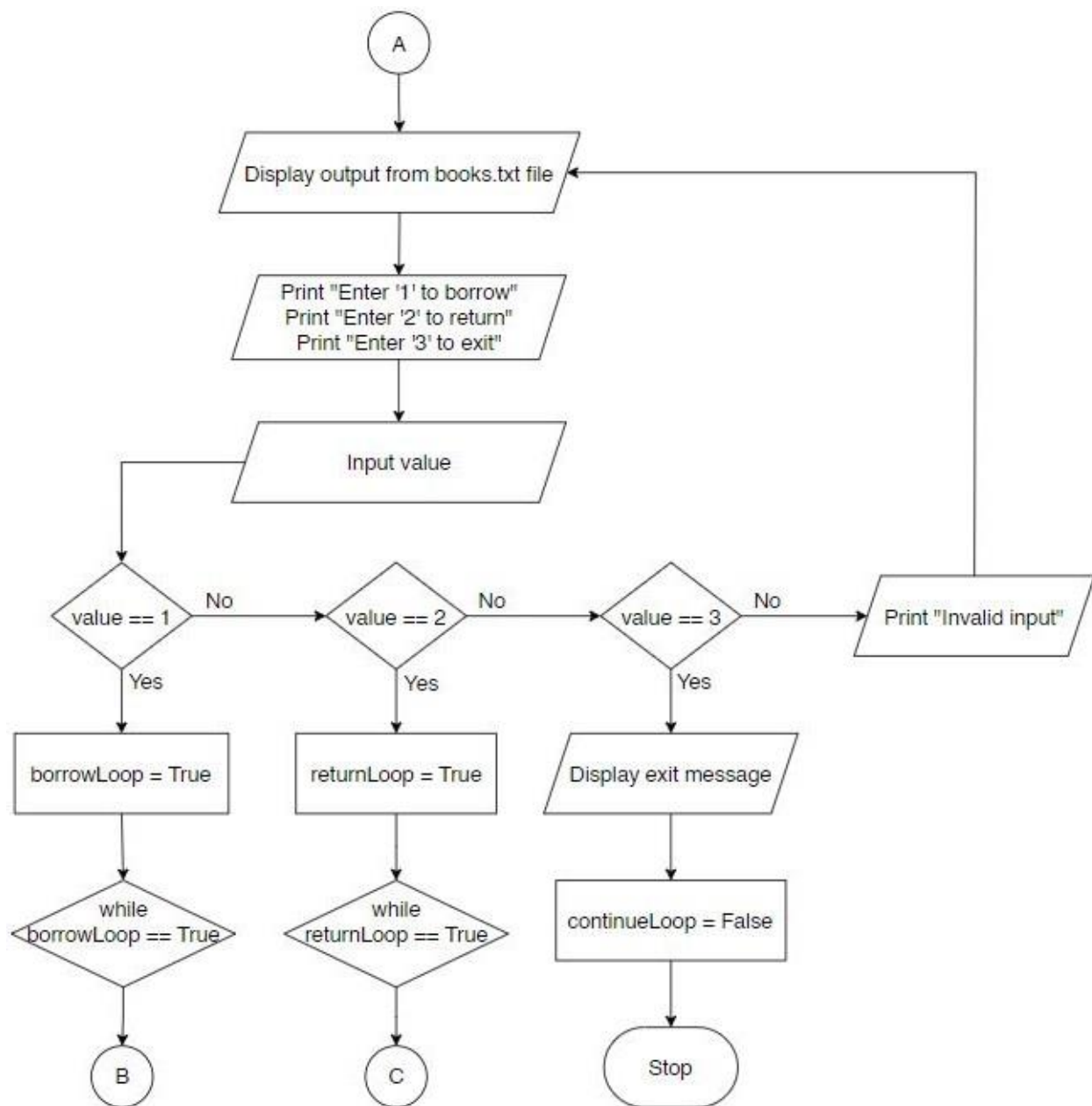


Figure 2: Flowchart of program from connector A

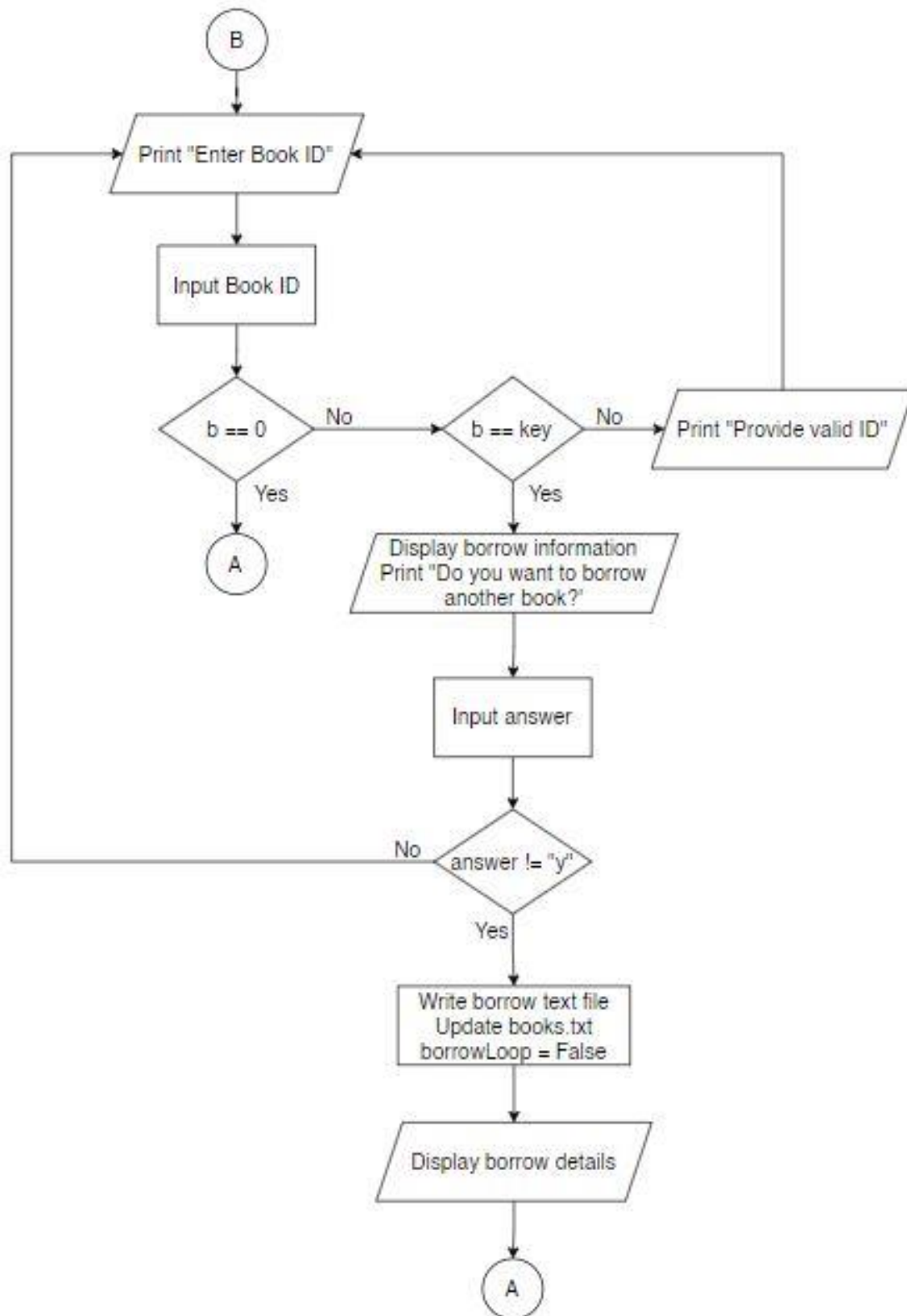


Figure 3: Flowchart of program from connector B

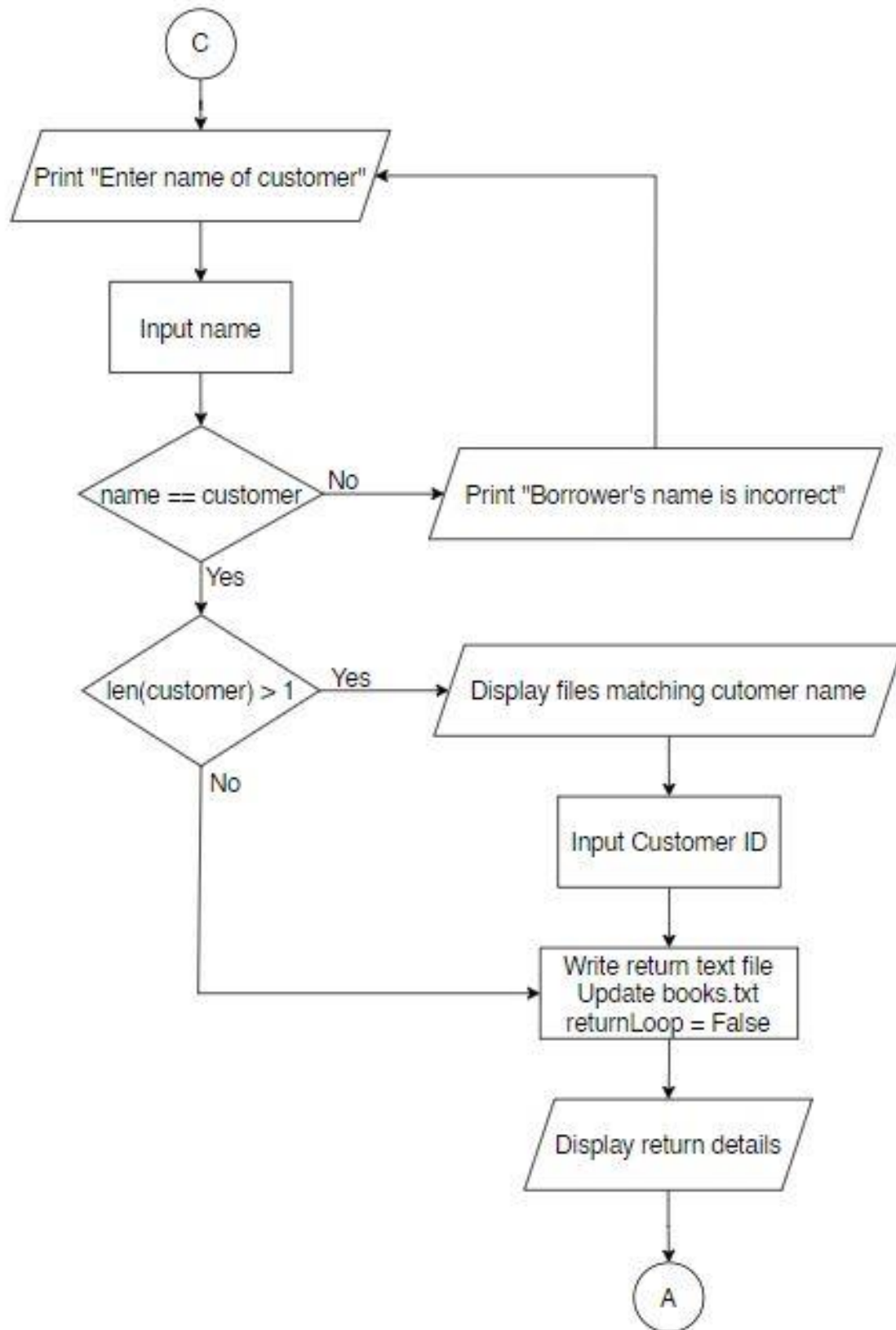


Figure 4: Flowchart of program from connector C

2.3 Pseudocode

A pseudocode is an unofficial way of coding description which does not need any programming language syntax or semantics. It is used for developing an outline of a program to understand the methods used in it. It is not an actual programming language so it cannot be compiled. It only summarizes the programs methods. (The Economic Times, 2021)

2.3.1 main.py

IMPORT modules

DECLARE FUNCITON

 INITIALIZE continueLoop as True

 WHILE continueLoop is equal to True

 TRY

 CALL FUNCTION to display book information from borrow module

 PRINT "Enter 1 to borrow"

 PRINT "Enter 2 to return"

 PRINT "Enter 3 to exit"

 PRINT "Select a value"

 INITIALIZE value by taking input as number from user

 IF number is equal to 1

 CALL borrow_book function from borrow module

 END IF

 ELIF number is equal to 2

 CALL return_book function from return module

 END ELIF

 ELIF number is equal to 3

 SET continueLoop to False

 Display exit message

 END ELIF

 ELSE

```
                PRINT invalid input message
            END ELSE
        END TRY
    EXCEPT
        PRINT invalid input message
    END EXCEPT
END WHILE
END FUNCTION
CALL FUNCTION
```

2.3.2 book_borrow.py

```
IMPORT MODULES
DECLARE FUNCTION
    OPEN books.txt file in reading mode
    INITIALIZE booksDictionary as an empty dictionary
    INITIALIZE bookID as 0
    FOR each line in file
        REPLACE new line with empty string
        INCREASE bookID by 1
        ASSIGN bookID as key and values as a list of each line
    END FOR
    CLOSE file
    RETURN booksDictionary
END FUNCTION
INITIALIZE booksDictionary as global vairable
DECLARE FUNCTION
    PRINT book information
    FOR all the keys and value in dictionary
        INITIALIZE values form dictionary separated by tab space
        PRINT keys and values
```

```
END FOR
END FUNCTION
DECLARE FUNCTION
    INITIALIZE borrowLoop as True
    INITIALIZE total to 0
    INITIALIZE books as empty list
    WHILE borrowLoop is equal to True
        TRY
            IF total is equal to 0
                PRINT "Press 0 to go back"
            END IF
            PRINT "Enter book ID"
            INITIALIZE b by taking number as input from user
            FOR all the keys and values in dictionary
                IF input is equal to key
                    INITIALIZE book as the name of book in dictionary
                    INITIALIZE quantity to the quantity of books in
                    dictionary as integer type
                    INITIALIZE price to the price of book in the dictionary
                    as float type
                    IF quantity is more than 0
                        PRINT "Book is available"
                        APPEND book to books
                        IF length of books is equal to 1
                            INITIALIZE borrower by taking string as
                            input from user
                            PRINT date from functions module
                            PRINT time from functions module
                        END IF
                        IF length of books is more than 1
                            PRINT price
```



```
INITIALIZE total as total plus price
OPEN books.txt file in write mode
FOR values in dictionary
    IF quantity is equal to values in
        second index
        INITIALIZE values in
            second index as integer
            values minus 1
        INITIALIZE values same
            values as string data type
    END IF
    WRITE all the index values in
        dictionary to books.txt file
END FOR
CLOSE file
CALL FUNCTION to display updated
book information
PRINT "Do you want to borrow another
book?"
INPUT answer as string value from user
IF answer is not equal to y
    SET borrowLoop to False
    INITIALIZE customer as borrower
        plus unique number from
        functions module and add .txt as
        string
    OPEN customer in write mode
    CALL date from functions module
        and write into file
    CALL time from function module
        and write into file
```

```
        WRITE borrower's name
        FOR values in books
            EXTRACT the index
            values of books and write
            into file
        END FOR
        CONVERT total into string type
        and write into file
        CLOSE file
        OPEN customer file in read mode
        FOR each line in file
            REPLACE new line with
            empty string
            PRINT line
        END FOR
        CLOSE file
    END IF
END IF
ELSE
    PRINT book not available message
END ELSE
ELIF b is equal to 0
    SET borrowLoop to False
    SET the loop to break
END ELIF
ELIF b is greater than length of booksDictionary OR b
is less than 0
    PRINT "Provide valid ID"
    SET the loop to break
END ELIF
```

```
        END IF
    END FOR
END TRY
EXCEPT
    PRINT "Provide valid ID"
END EXCEPT
END WHILE
END FUNCTION
```

2.3.3 book_return.py

```
IMPORT modules
DECLARE FUNCTION
    INITIALIZE returnLoop as True
    WHILE returnLoop is equal to True
        TRY
            INITIALIZE name by taking string as input from user
            INITIALIZE customer by taking name and use module to search the
            matching filenames in the directory and extract them in list
            IF length of customer is equal to 1
                INITIALIZE returner as value from 0 index of customer
            END IF
            ELIF length of customer is greater than 1
                INITIALIZE idLoop as True
                WHILE idLoop is equal to True
                    TRY
                        PRINT "Customer ID" and "filename"
                        FOR values in customer
                            INITIALIZE n as index value plus 1
                            PRINT n as string and the values from
                                customer
```

```
        END FOR
        PRINT "Select the customer ID"
        INITIALIZE c by taking input from user as
        integer type
        IF c is greater than length customer or less
        than 1
            PRINT "Provide valid id"
        END IF
        ELSE
            INITIALIZE r as c minus 1
            INITIALIZE returner as customer value
            in r index
            SET idLoop to False
        END ELSE
    END TRY
    EXCEPT
        PRINT "Provide valid ID"
    END EXCEPT
END WHILE
END ELIF
PRINT "Borrow details"
INITIALIZE books as empty list
OPEN returner in reading mode
INITIALIZE lines the values from each line in file
FOR each line in lines
    INITIALIZE line as value from each line and replace new line
    with empty string
    APPEND line in books
    PRINT line
END FOR
CLOSE file
```

```
INITIALIZE returned as name plus unique number from functions
module and add .txt as string
OPEN returned in write mode
CALL date value from functions module
CALL time value from functions module
WRITE name of customer into file
FOR i in range 6 to length of books
    WRITE the values of books in i index into file
END FOR
IMPORT module to work with date and time
INITIALIZE borrowDate as empty list
APPEND date value from 0 index of books to borrowDate
INITIALIZE startDate as value in 0 index of borrowDate in date
format
INITIALIZE dateToday as date value from functions module in date
format
INITIALIZE endDate as startDate plus 10 days
IF dateToday > endDate
    INITIALIZE days as dateToday mins endDate in day format
    INITIALIZE fine as 0.25 times days
    INITIALIZE total as float number value of ninth index in
    books and remove other text
    INITIALIZE grand the sum of fine and total
    WRITE fine as string into file
    WRITE grand as string into file
    WRITE days as string into file
END IF
CLOSE file
OPEN returned in read mode
PRINT "The book has been returned"
FOR each line in file
```

```
        INITIALIZE line as values from each line of file and remove
        new line with empty string
        PRINT line
    END FOR
    CLOSE file
    FOR i in range 6 and length of books minus 2
        OPEN books.txt file in write mode
        FOR values in booksDictionary of book_borrow module
            IF index value of books is equal to values in 0 index
                INITIALIZE values of second index as integer
                values of second index plus 1
                INITIALIZE values of second index as string
                values
            END IF
            WRITE all the index values of dictionary into books.txt
            file
        END FOR
        CLOSE file
    END FOR
    SET returnLoop to False
END TRY
EXCEPT
    PRINT "Borrower's name is incorrect"
END EXCEPT
END WHILE
END FUNCTION
```

2.3.4 functions.py

```
DECLARE FUNCTION
    IMPORT datetime module
```

```
    INITIALIZE date as current date in string format
    INITIALIZE time as current time in string format
    INITIALIZE unique as combination of date and time in string format
    RETURN date, time and unique
END FUNCTION
DECLARE date, time and unique as global variable
```

2.3.5 messages.py

```
IMPORT book_borrow module
DECLARE FUNCTIONS for displaying various messages as required
END FUNCTIONS
```

2.4 Data Structures

Data structures are the tools which provide a method to organize and store data so that they can be accessed and used effectively. They establish the connection between the data and the functions that can be performed on them. (Jaiswal, 2017)

There are various data structures in python. Some of the primitive data structures are integers, float, strings and boolean. Likewise, some complex data structures are lists, tuples, dictionary and sets. These data structures are used to perform various operations for input/output, data storage, character and string processing, etc.

Some of the data structures used in the program are as follows:

2.4.1 Integer

An integer represents whole numbers or numeric data ranging from negative to positive infinity. This data structure can be used in python by two ways one is by directly assigning it to a variable or by using the built-in keyword “int” for converting the numeric data into integer type data. Both of these methods have been used while developing the program and its implementation is presented in the figure below.

```
total = 0
books = []
while borrowLoop == True:
    try:
        if total == 0:
            print("Press '0' to go back.")
            print()
            b = int(input("Enter the ID of the book you want to borrow: "))
```

Figure 5: Integer data structure in python

2.4.2 String

A string is a group of alphabets, words or other characters. It can be used in python by either placing the alphabets or character inside a double quotation or by using the built-in keyword str to convert any character into string data type. Both of these methods have been used while developing the program and its implementation is presented in the figure below.


```
file.write(str(values[0]) + "," + str(values[1]) + "," + str(values[2]) + "," + str(values[3]) + "\n")
file.close()
display()
print("Do you want to borrow another book?")
answer = input("If 'Yes' enter 'y'. Press any other key to skip: ").lower()
```

Figure 6: String data structure in python

2.4.3 Boolean

Boolean refers to the binary value 0 or 1 which is interchangeable with true and false. This is useful mainly for comparison or providing a condition for variables. It can be used in python by assigning True or False to variables or by using the built-in keyword "bool" to get the value in boolean data type. This data structure has been used while developing the program and its implementation is presented in the figure below.

```
returnLoop = True
while returnLoop == True:
    try:
        name = input("Enter the name of customer: ")
```

Figure 7: Boolean data structure in python

2.4.4 Dictionary

A dictionary is an unordered collection of data in the form of key-value pairs where each key is unique. The keys and values are separated by colon and each pair is separated by comma. This data structure can be used in python by either using curly brackets "{}" or by using the built-in function "dict()". This data structure has been used while developing the program and its implementation is presented in the figure below.

```
file = open("books.txt", "r")
booksInDictionary = {}
bookID = 0
for line in file:
    line = line.replace("\n", "")
    bookID += 1
    booksInDictionary[bookID] = line.split(",")
file.close()
return booksInDictionary
```

Figure 8: Dictionary data structure in python

2.4.5 List

A list is a collection data type which stores elements in an ordered sequence and can be used to store heterogeneous items. This data structure can be used in python by either using square brackets "[]" or by using the built-in function "list()". This data structure has been used while developing the program and its implementation is presented in the figure below.

```
borrowDate = []
borrowDate.append(books[0].replace("Date: ", "")) #Extracts time from borrow file
```

Figure 9: List data structure in python

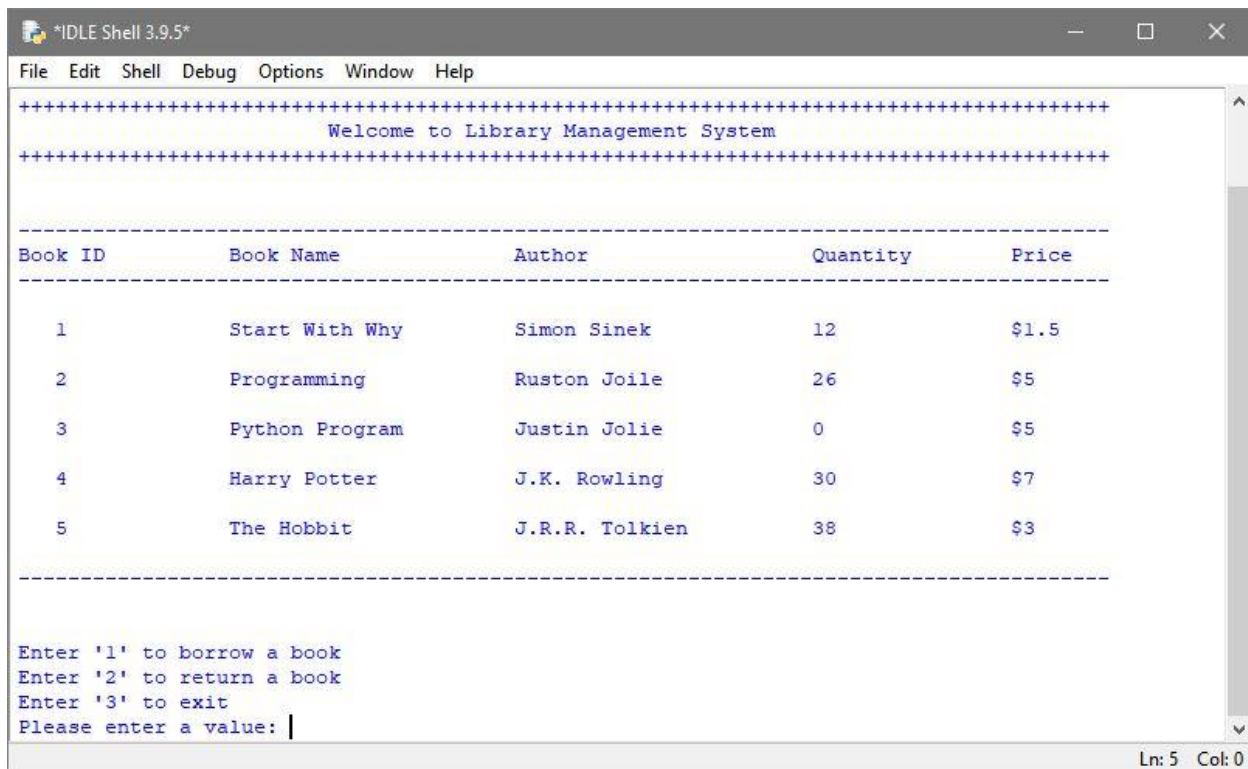
3. Program

A program is a set of methods and instructions which helps to manipulate a given data by taking input and processing it as required to get the desired output. (Computer Hope, 2021)

In this project, a library management system was created to track the books borrowed and returned by the customers. The screenshot of all the processes involved is attached along with their description for all processes to demonstrate the working of the program.

All the processes involved from start to end of the program are elaborated below.

After starting the program, a welcome message is displayed to the user along with a list of all the books and their details present in stock. Then a selection message is displayed to the user instructing them to input a value as 1 to borrow books, 2 to return books or 3 to exit the program.



```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
+++++
Welcome to Library Management System
+++++

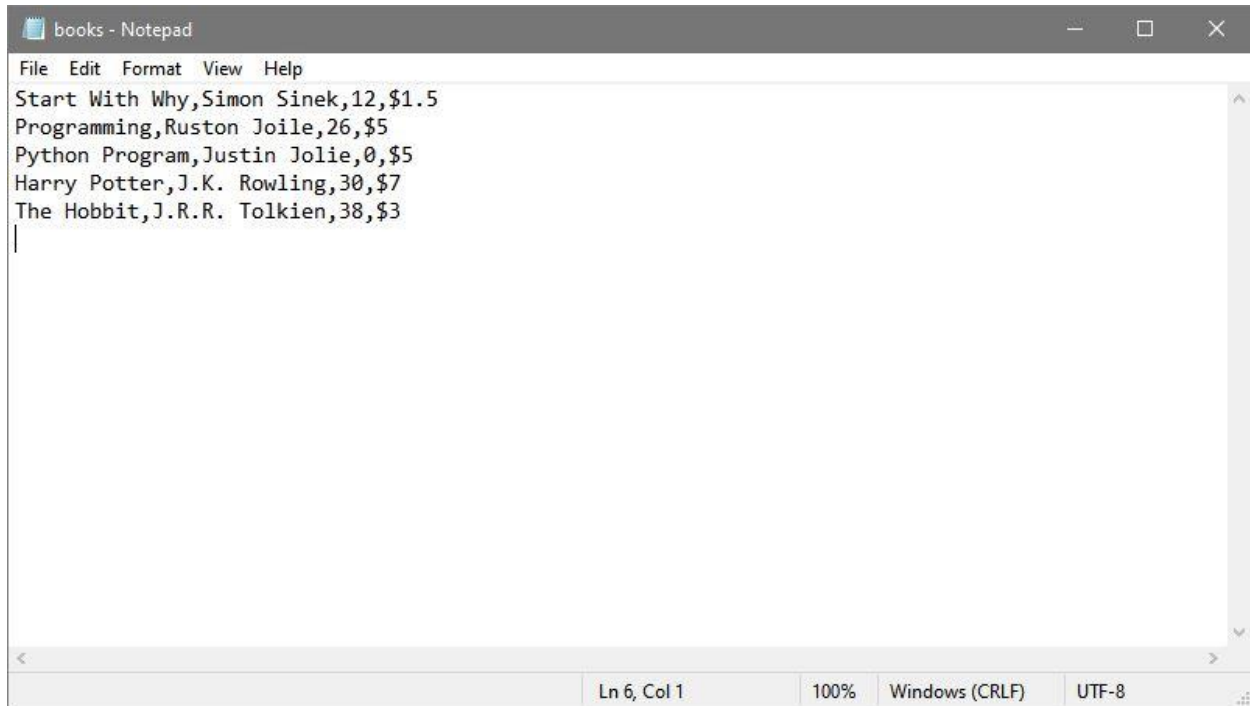
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek  12            $1.5
2           Programming     Ruston Joile  26            $5
3           Python Program  Justin Jolie  0             $5
4           Harry Potter    J.K. Rowling  30            $7
5           The Hobbit       J.R.R. Tolkien 38            $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
Ln: 5 Col: 0

```

Figure 10: User Interface after starting the program

The details present in the stock file is depicted below.



The image shows a Notepad window titled "books - Notepad". The window contains a text file with the following content:

```
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,26,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,30,$7
The Hobbit,J.R.R. Tolkien,38,$3
|
```

The status bar at the bottom indicates "Ln 6, Col 1", "100%", "Windows (CRLF)", and "UTF-8".

Figure 11: Stock file containing book details

If a string is entered as value instead of 1,2 or 3, then a message is displayed to the user notifying the user that an invalid input has been entered and guide them to provide a valid input. Then it again displays the book details and asks input from user.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: one

+++++
Invalid input
Please provide value as 1, 2 or 3.
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek  12            $1.5
2           Programming     Ruston Joile 26            $5
3           Python Program  Justin Jolie 0              $5
4           Harry Potter    J.K. Rowling 30            $7
5           The Hobbit      J.R.R. Tolkien 38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
  
```

Ln: 44 Col: 0

Figure 12: Output after receiving string input as value

Likewise, if any other number is entered as value instead of 1,2 or 3, then a message is displayed to the user notifying the user that an invalid input has been entered and guide them to provide a valid input. Then it again displays the book details and asks input from user.

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 5

+++++
Invalid input
Please provide value as 1, 2 or 3.
+++++

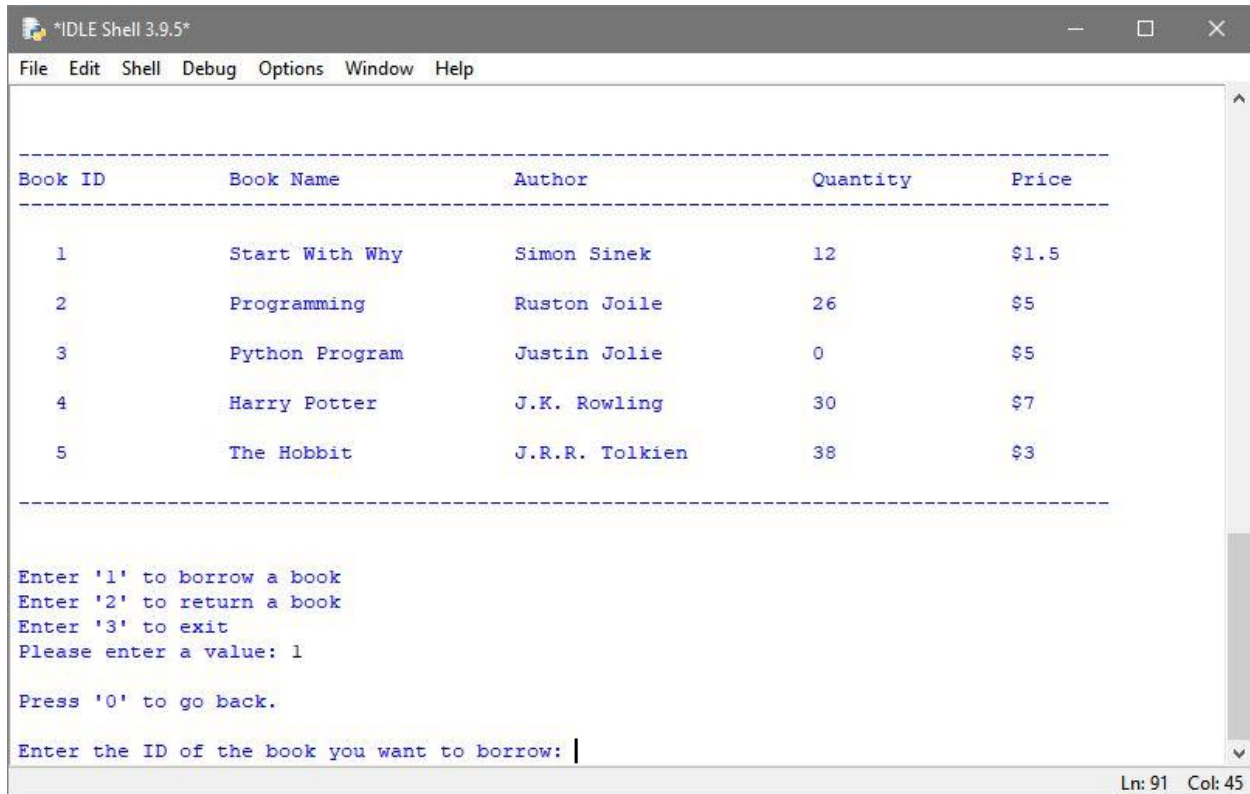
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie    0           $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
Ln: 71 Col: 7

```

Figure 13: Output after receiving unspecified number as value

If 1 is entered as input, it proceeds to the borrow part. A message is displayed to the user indicating them to press '0' to go back to the previous screen. Another message is displayed where it asks the user to input the book ID of the book they want to borrow.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity    Price
-----
1           Start With Why  Simon Sinek  12          $1.5
2           Programming     Ruston Joile 26          $5
3           Python Program  Justin Jolie 0           $5
4           Harry Potter    J.K. Rowling 30          $7
5           The Hobbit      J.R.R. Tolkien 38          $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 1

Press '0' to go back.

Enter the ID of the book you want to borrow: |
```

Ln: 91 Col: 45

Figure 14: Output after receiving 1 as value

If a book ID is entered where quantity of the books is 0, the program displays a message notifying the user that the given book is not available. Again, the program displays the book details and asks the user to provide a book ID.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Enter the ID of the book you want to borrow: 3

+++++
                        Book is not available
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why   Simon Sinek    12           $1.5
2           Programming      Ruston Joile   26           $5
3           Python Program    Justin Jolie   0            $5
4           Harry Potter      J.K. Rowling   30           $7
5           The Hobbit        J.R.R. Tolkien 38           $3
-----

Press '0' to go back.
Enter the ID of the book you want to borrow: |
Ln: 243 Col: 0

```

Figure 15: Output after receiving book ID as input where quantity is 0

If a string value is entered as book ID, it displays a message alerting the user to provide a valid ID and again the program displays the book details and asks the user to input a valid book ID.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Press '0' to go back.
Enter the ID of the book you want to borrow: one
+++++
                        Please provide a valid ID
+++++

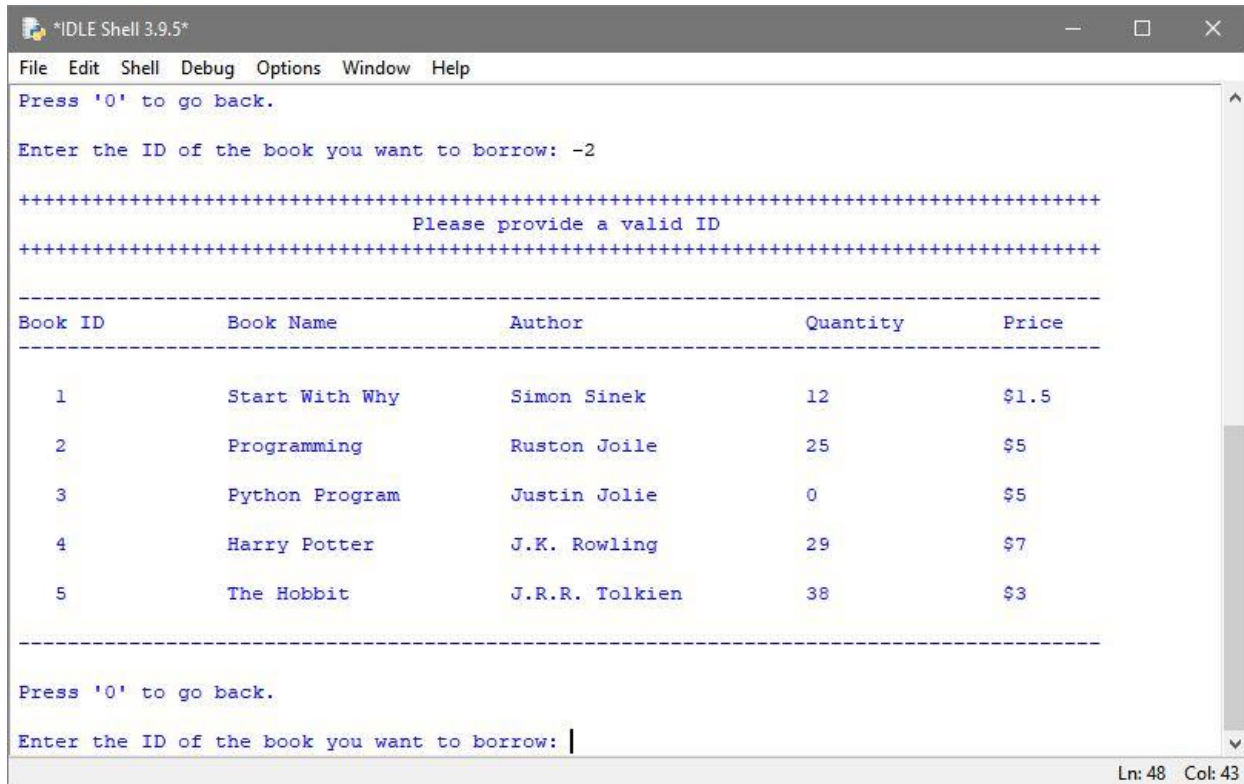
-----
Book ID      Book Name      Author      Quantity    Price
-----
1           Start With Why   Simon Sinek  12           $1.5
2           Programming         Ruston Joile  26           $5
3           Python Program       Justin Jolie  0            $5
4           Harry Potter          J.K. Rowling  30           $7
5           The Hobbit             J.R.R. Tolkien 38           $3
-----

Press '0' to go back.
Enter the ID of the book you want to borrow: |
Ln: 289 Col: 0

```

Figure 16: Output after receiving string input as book ID

If any numeric data is entered which does not match with the displayed book ID, it gives a message notifying the user to provide a valid ID. Again, the program displays the book details and asks the user to provide a book ID.



```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Press '0' to go back.
Enter the ID of the book you want to borrow: -2

+++++
                        Please provide a valid ID
+++++

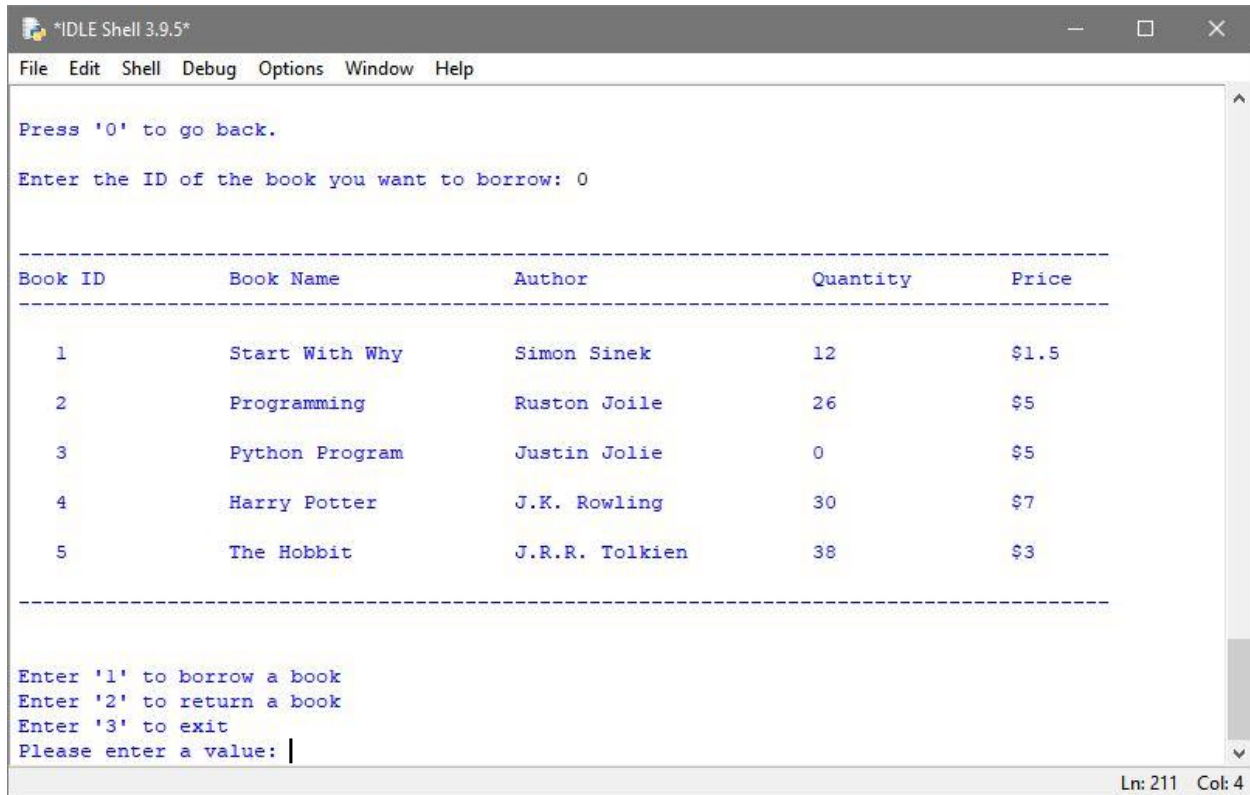
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why   Simon Sinek    12           $1.5
2           Programming      Ruston Joile   25           $5
3           Python Program   Justin Jolie   0            $5
4           Harry Potter     J.K. Rowling   29           $7
5           The Hobbit        J.R.R. Tolkien 38           $3
-----

Press '0' to go back.
Enter the ID of the book you want to borrow: |
Ln: 48 Col: 43

```

Figure 17: Output after receiving unspecified number as book ID

If 0 is entered as book ID, the program returns back previous screen and asks the user to input to enter 1 to borrow, 2 to return or 3 to exit.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

Press '0' to go back.
Enter the ID of the book you want to borrow: 0

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
```

Ln: 211 Col: 4

Figure 18: Output after receiving 0 as book ID

If a book ID is entered which matches with the given ID and the quantity of books is more than 0, it displays a message notifying the user that the book is available. Then, it asks the user to enter the name of borrower.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity    Price
-----
1           Start With Why  Simon Sinek  12          $1.5
2           Programming     Ruston Joile 26          $5
3           Python Program  Justin Jolie 0           $5
4           Harry Potter    J.K. Rowling 30          $7
5           The Hobbit      J.R.R. Tolkien 38          $3
-----

Press '0' to go back.

Enter the ID of the book you want to borrow: 4

+++++
                        Book is available
+++++

Enter the name of Borrower: |
Ln: 307 Col: 28

```

Figure 19: Output after receiving appropriate book ID as input

After the name has been entered, the program displays the details such as the date and time of borrow and the price of the given book. Also, it updates the quantity of the borrowed book. Then, it displays the updated book details again and asks whether the user wants to borrow another book. Another message is displayed asking the user to enter “y” if they want to borrow more books, if not then any other key can be pressed to skip to the billing part.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

+++++
                        Book is available
+++++

Enter the name of Borrower: John
The date of borrow is: 01-09-2021
The time of borrow is: 10:37:14 AM
The price of the book is: $7.0

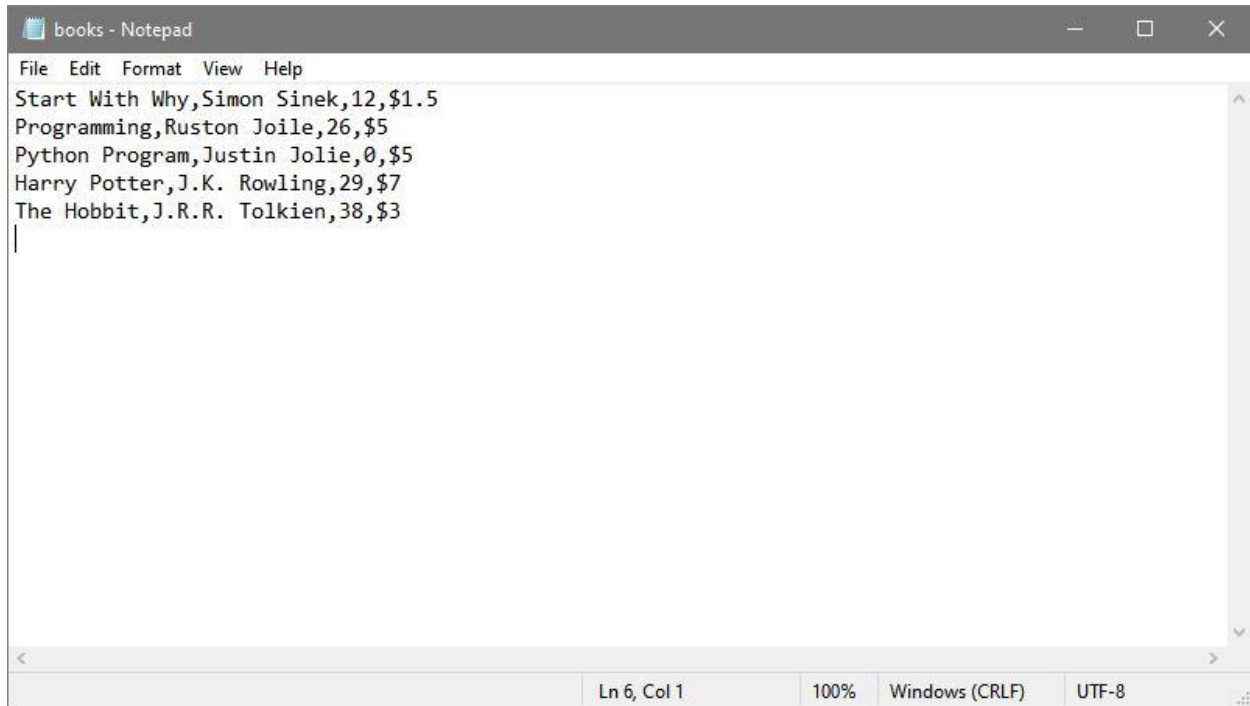
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek    12           $1.5
2           Programming      Ruston Jolle   26           $5
3           Python Program   Justin Jolie    0           $5
4           Harry Potter     J.K. Rowling   29           $7
5           The Hobbit       J.R.R. Tolkien  38           $3
-----

Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: |
Ln: 74 Col: 0

```

Figure 20: Output after receiving borrower's name

The updated details in the stock file after a book has been borrowed is depicted below.



```
books - Notepad
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,26,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,29,$7
The Hobbit,J.R.R. Tolkien,38,$3
|
Ln 6, Col 1 100% Windows (CRLF) UTF-8
```

Figure 21: Stock file after a book is borrowed

If “y” or “Y” is entered, the program asks the user to enter the ID of the book which they want to additionally borrow.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Book is available
+++++
Enter the name of Borrower: John
The date of borrow is: 01-09-2021
The time of borrow is: 10:37:14 AM
The price of the book is: $7.0

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: y

Enter the ID of the book you want to borrow: |
Ln: 90 Col: 45

```

Figure 22: Output after receiving 'y' as input to borrow another book

If a book ID is entered which matches with the given ID and the quantity of books is more than 0, it displays a message notifying the user that the book is available. Then, displays the price of the book. Also, it updates the quantity of the borrowed book and displays the updated book details for the remaining books. Again, it asks the user whether they want to borrow another book.

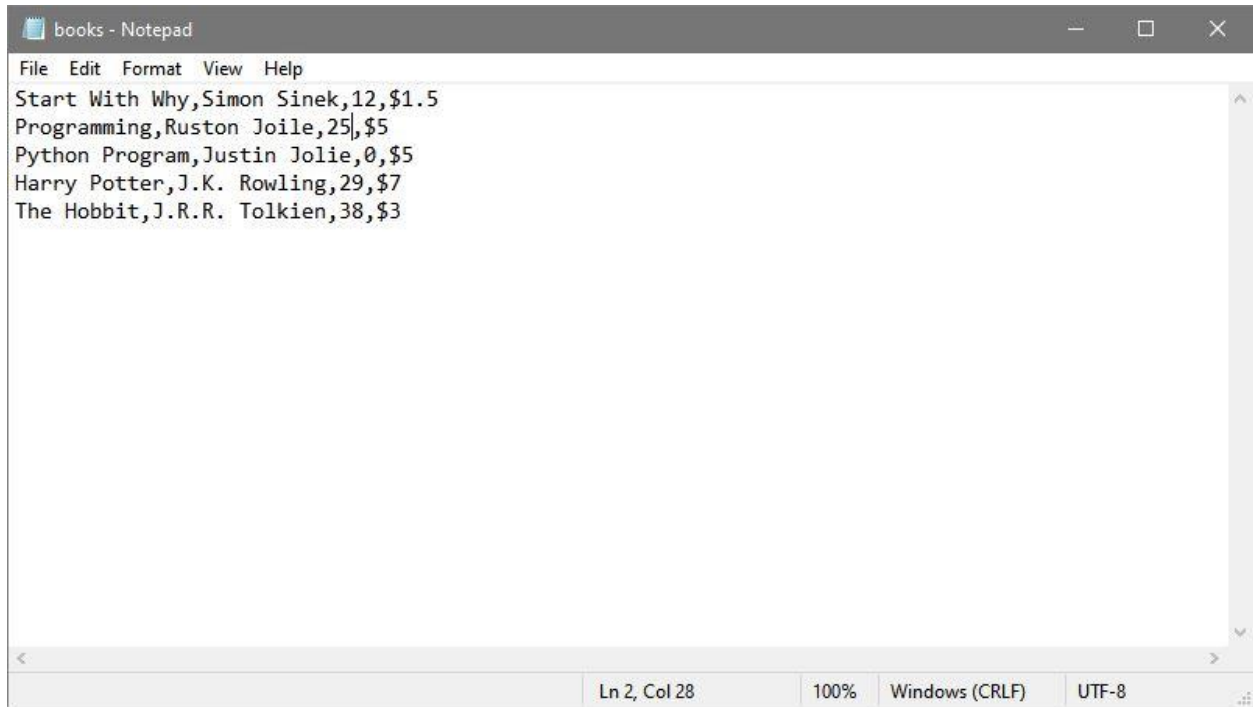
```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
If 'Yes' enter 'y'. Press any other key to skip: y
Enter the ID of the book you want to borrow: 2
+++++
                        Book is available
+++++
The price of the book is: $5.0
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek    12           $1.5
2           Programming         Ruston Joile   25           $5
3           Python Program      Justin Jolie   0            $5
4           Harry Potter        J.K. Rowling   29           $7
5           The Hobbit          J.R.R. Tolkien 38           $3
-----
Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: |
Ln: 106 Col: 0

```

Figure 23: Output after borrowing another book

The updated details in the stock file after another book has been borrowed is depicted below.



```
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,25,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,29,$7
The Hobbit,J.R.R. Tolkien,38,$3
```

Ln 2, Col 28 100% Windows (CRLF) UTF-8

Figure 24: Stock file after another book is borrowed

If any value is entered other than “y” or “Y”, the program displays all the details of the transaction. Then, it returns to the main screen and displays updated book details of the books after the borrow transaction and asks the user to input values 1,2 or 3 to borrow, return or exit.

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: n

Date: 01-09-2021
Time: 10:37:14 AM

Customer's Name: John

Books borrowed:
Harry Potter
Programming

Total price: $12.0
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   25           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien 38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
Ln: 131 Col: 0

```

Figure 25: Output after borrow completion

The bill generated for the customer after they have borrowed the books which must be paid at the time of returning is depicted below.

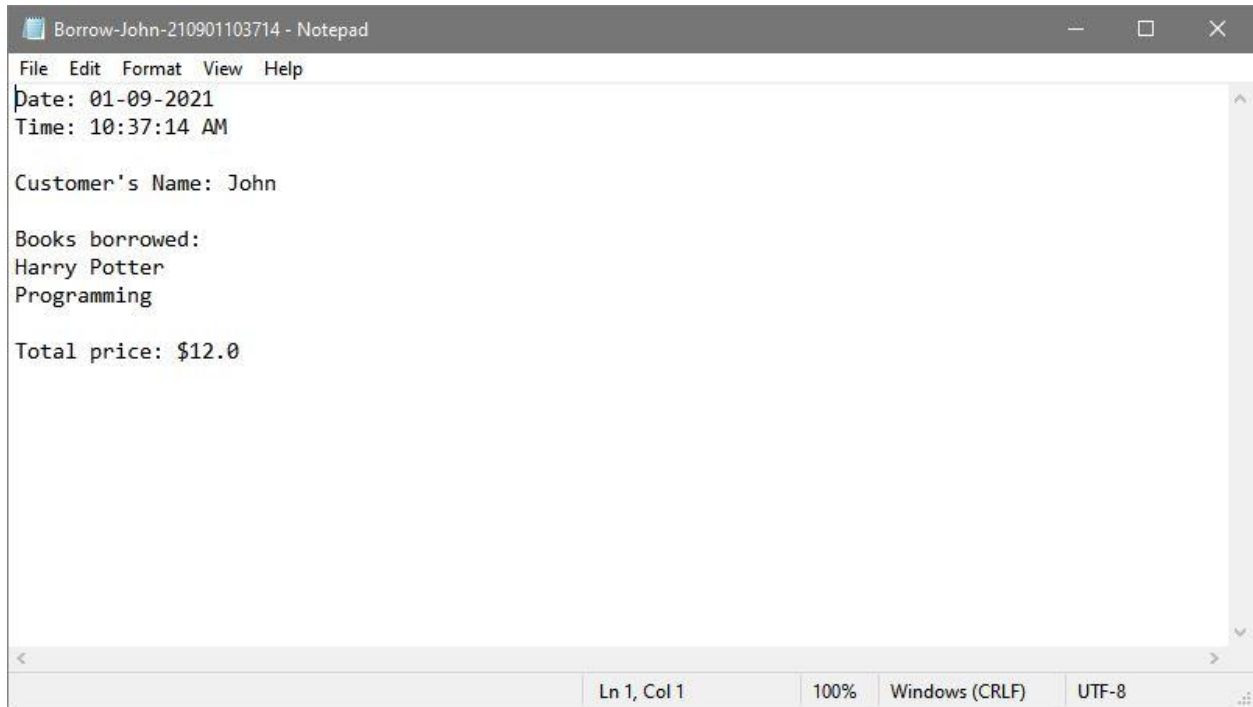
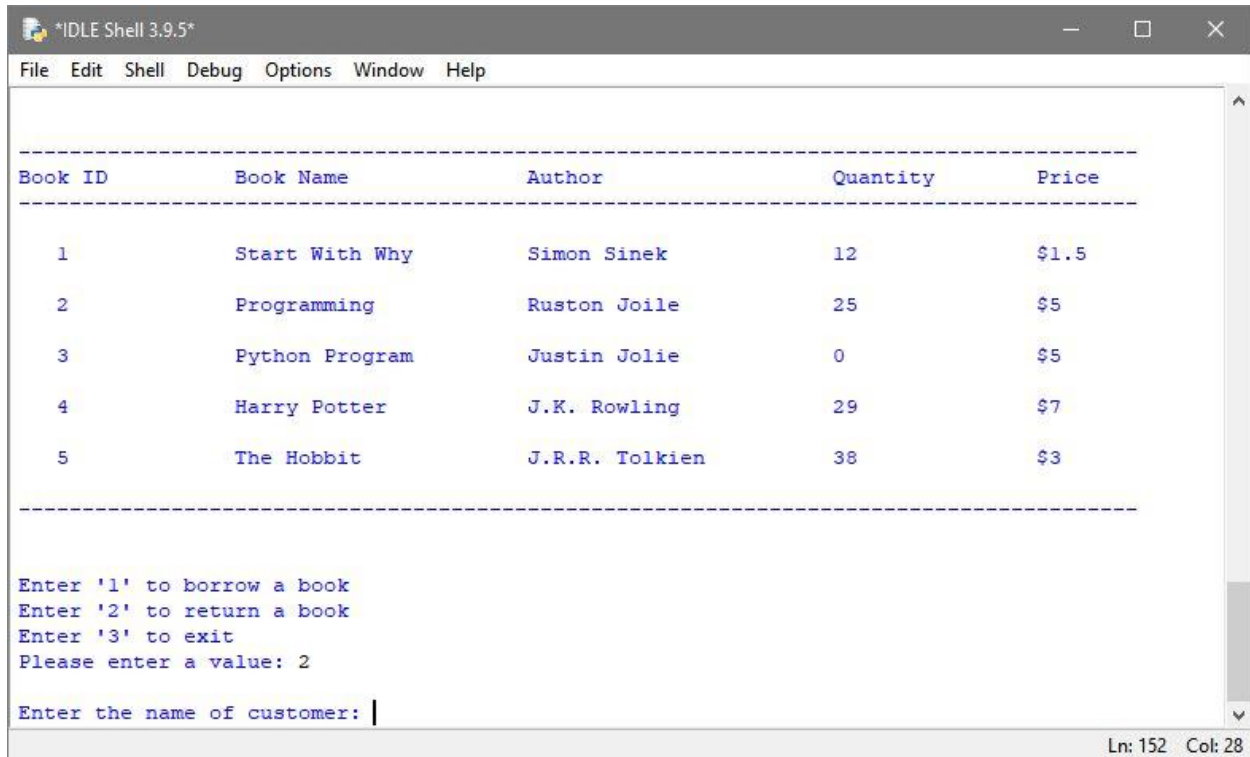


Figure 26: Bill generated after borrowing the books

If 2 is entered as value, the program proceeds to the return part. Then it asks the user to input the name of customer who wants to return the book.



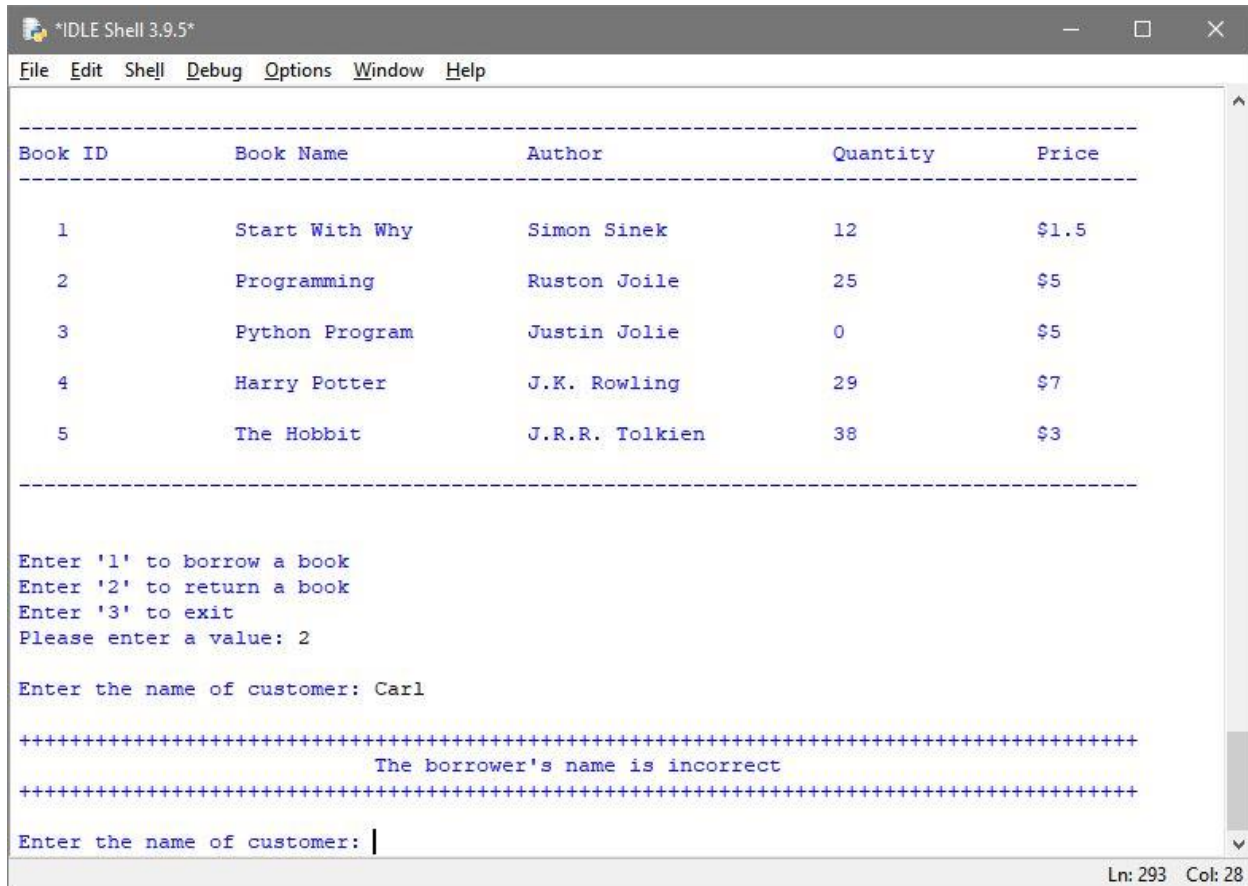
The screenshot shows a Python IDLE Shell window with a menu bar (File, Edit, Shell, Debug, Options, Window, Help) and a text area containing a table of books and user prompts. The table has five columns: Book ID, Book Name, Author, Quantity, and Price. Below the table, there are three prompts: 'Enter '1' to borrow a book', 'Enter '2' to return a book', and 'Enter '3' to exit'. The user has entered '2' in response to the 'Please enter a value:' prompt. The prompt 'Enter the name of customer:' is followed by a cursor. The status bar at the bottom right indicates 'Ln: 152 Col: 28'.

Book ID	Book Name	Author	Quantity	Price
1	Start With Why	Simon Sinek	12	\$1.5
2	Programming	Ruston Joile	25	\$5
3	Python Program	Justin Jolie	0	\$5
4	Harry Potter	J.K. Rowling	29	\$7
5	The Hobbit	J.R.R. Tolkien	38	\$3

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2
Enter the name of customer: |

Figure 27: Output after receiving 2 as input in value

If a name is entered which does not match the name of the borrower, it displays a message alerting the user that the borrower's name is incorrect. Then it again asks the user to enter the name of the customer who wants to return the book.



```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek      12           $1.5
2           Programming        Ruston Joile     25           $5
3           Python Program    Justin Jolie     0            $5
4           Harry Potter      J.K. Rowling     29           $7
5           The Hobbit        J.R.R. Tolkien   38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: Carl

+++++
                        The borrower's name is incorrect
+++++

Enter the name of customer: |
Ln: 293 Col: 28

```

Figure 28: Output after receiving unknown customer name

If the name of the customer is entered which matches the name of borrower, the details for the books borrowed by that person is displayed. If the books have been returned within the lending period (i.e., 10 days), a message is displayed notifying the user that the books have been returned and displays the details return transaction.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Enter the name of customer: John

Borrow Details:

Date: 01-09-2021
Time: 10:37:14 AM

Customer's Name: John

Books borrowed:
Harry Potter
Programming

Total price: $12.0

+++++
                        The book has been returned
+++++

Date: 05-09-2021
Time: 10:55:54 AM

Returned By: John

Books returned:
Harry Potter
Programming

Total price: $12.0
+++++
Ln: 440 Col: 0
```

Figure 29: Output after receiving valid customer name

The bill generated for the customer after returning the book is depicted below.

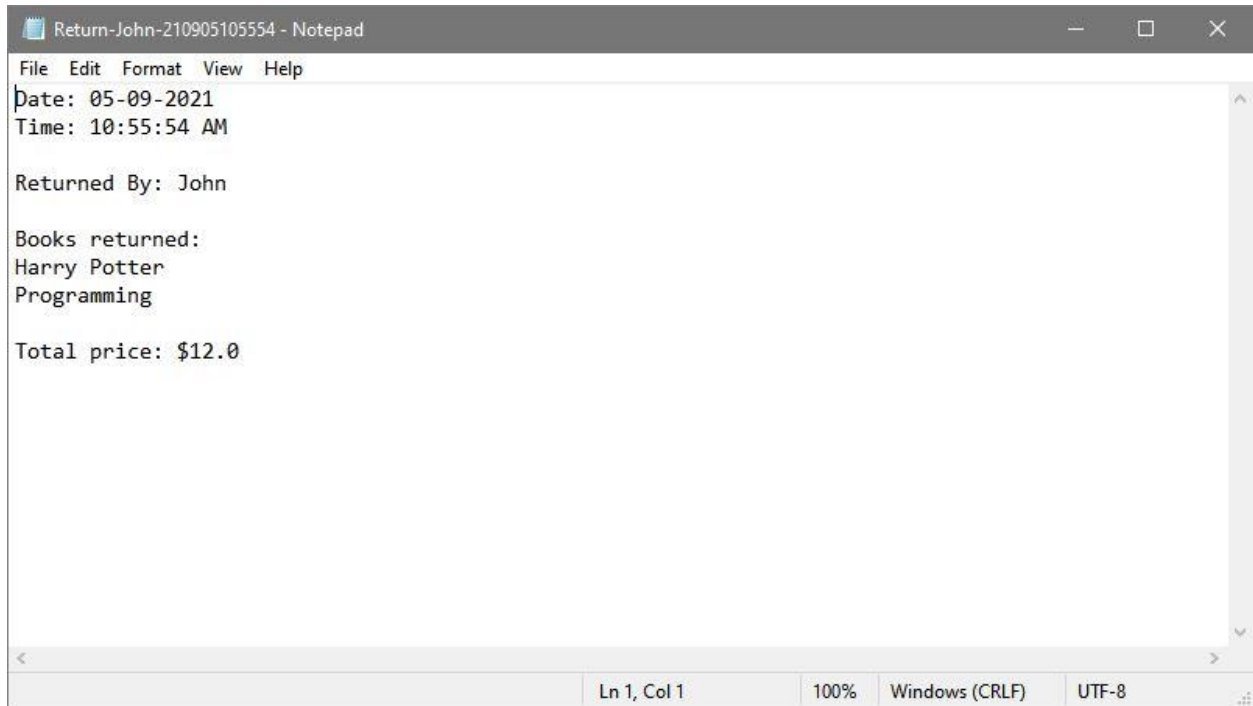
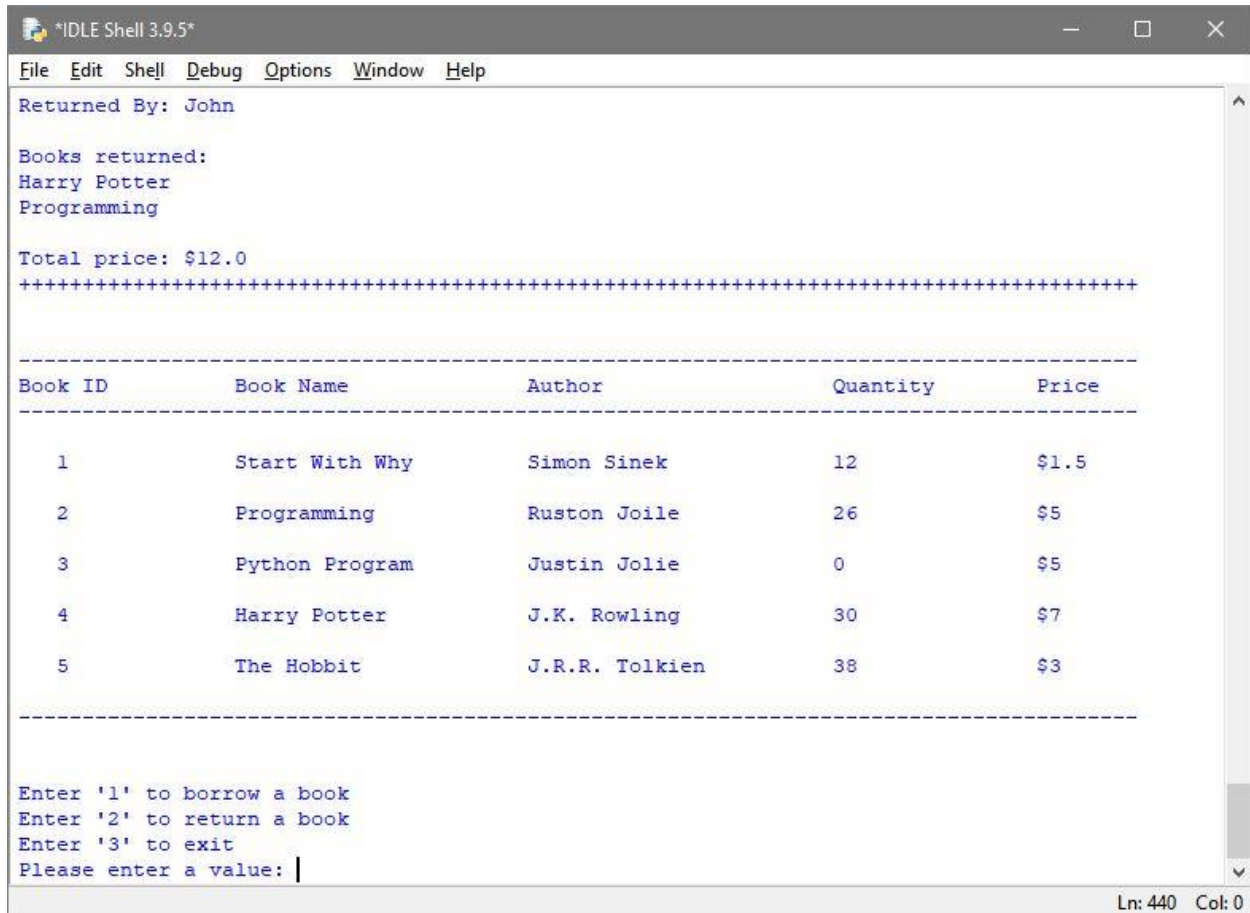


Figure 30: Bill generated after returning the books

Then, it displays the updated book details after the books which have been returned. After that, it again goes back to the previous screen and asks the user to input 1,2 or 3 to borrow, return or exit.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Returned By: John

Books returned:
Harry Potter
Programming

Total price: $12.0
+++++

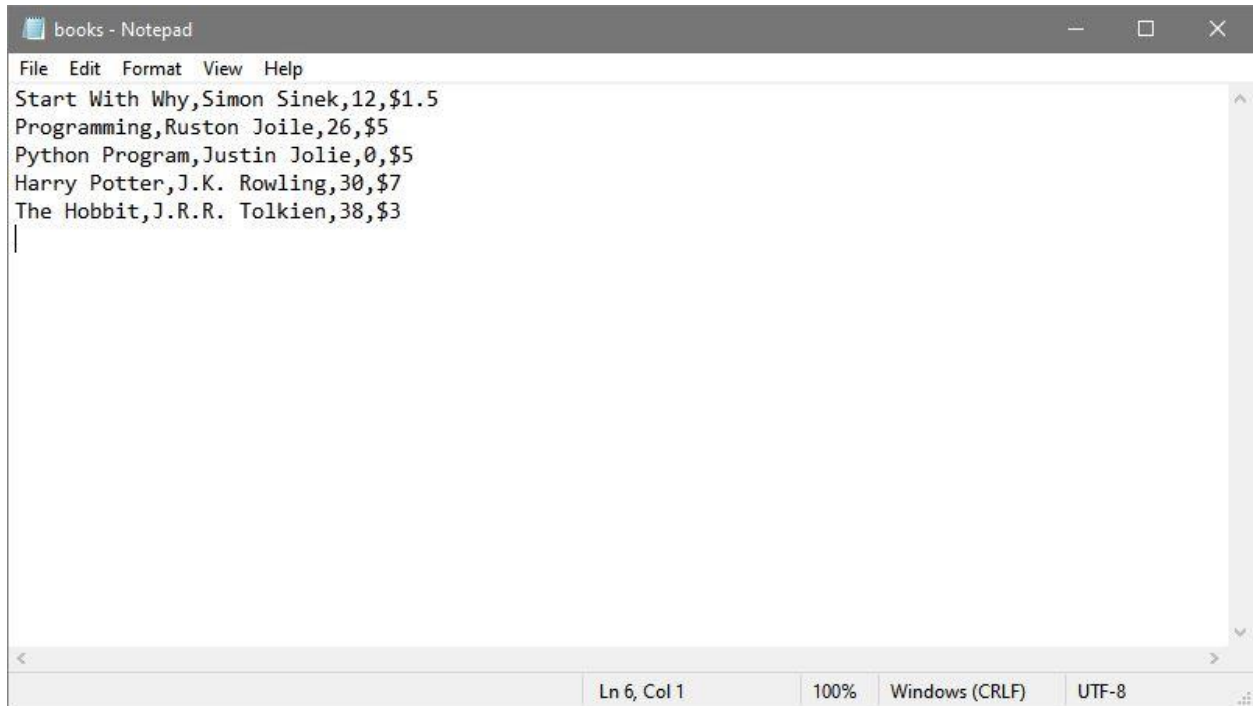
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie    0           $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
```

Ln: 440 Col: 0

Figure 31: Output after receiving valid customer name continued

The updated details in the stock file after a book has been returned is depicted below.

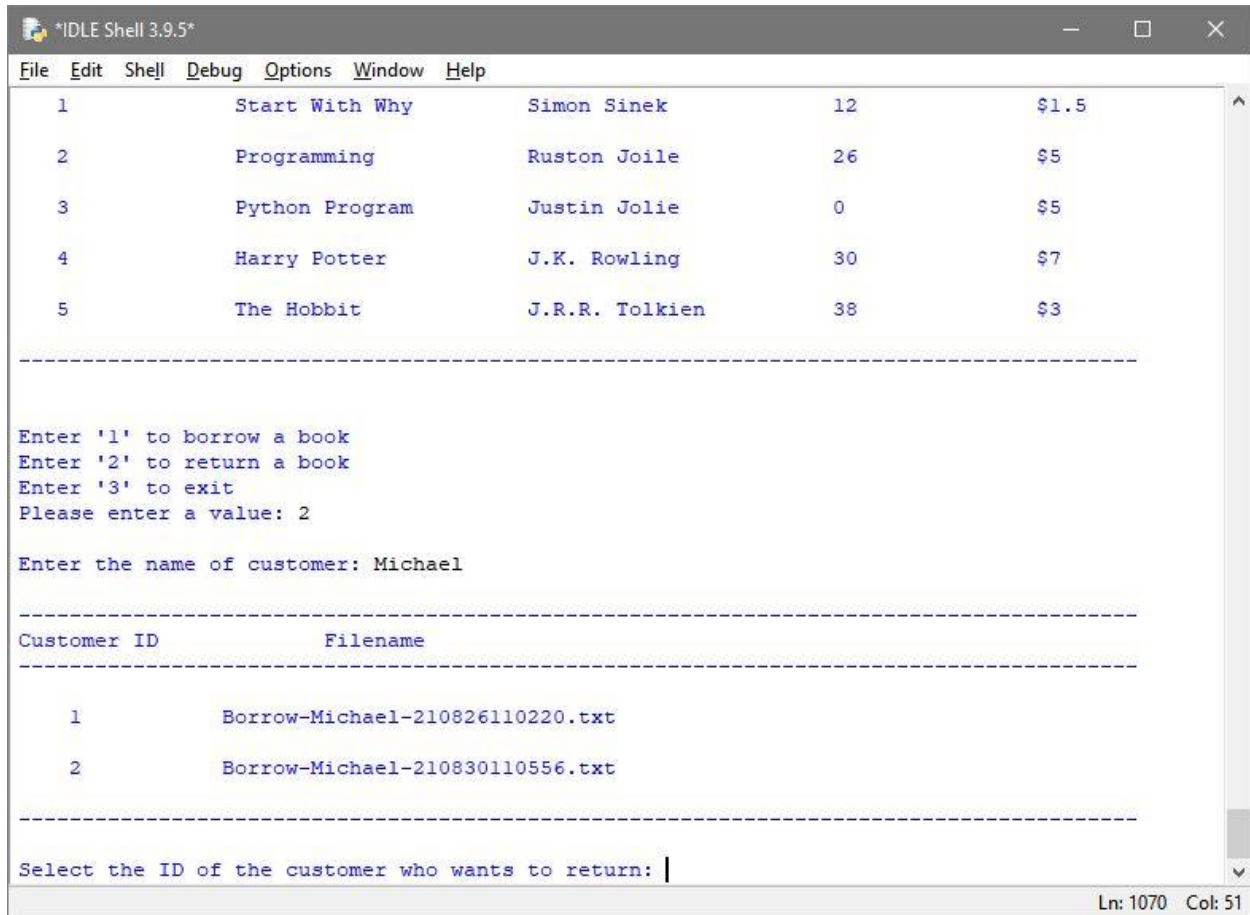


```
books - Notepad
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,26,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,30,$7
The Hobbit,J.R.R. Tolkien,38,$3
|
```

Ln 6, Col 1 100% Windows (CRLF) UTF-8

Figure 32: Stock file after returning the books

If a name has been entered, which matches the name of multiple borrowers, the list of customers with the same name is displayed. Then, it asks the user to enter the ID of the customer who wants to return the book.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
1      Start With Why      Simon Sinek      12      $1.5
2      Programming      Ruston Joile      26      $5
3      Python Program      Justin Jolie      0      $5
4      Harry Potter      J.K. Rowling      30      $7
5      The Hobbit      J.R.R. Tolkien      38      $3

-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: Michael

-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt
-----

Select the ID of the customer who wants to return: |
Ln: 1070 Col: 51
```

Figure 33: Output when multiple customer name is same

If a string value is entered as customer ID, it displays a message alerting the user to provide a valid ID. Again, it displays the list of customers and asks the user to enter the ID of the particular customer who wants to return the books.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

Enter the name of customer: Michael

-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt
-----

Select the ID of the customer who wants to return: two

+++++
                        Please provide a valid ID
+++++

-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt
-----

Select the ID of the customer who wants to return: |
```

Ln: 1363 Col: 51

Figure 34: Output when string input is received as customer ID

Likewise, if a number value is entered which does not match the given customer ID, it displays a message alerting the user to provide a valid ID. Again, it displays the list of customers and asks the user to enter the ID of the particular customer who wants to return the books.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt
-----

Select the ID of the customer who wants to return: -1

+++++
                        Please provide a valid ID
+++++

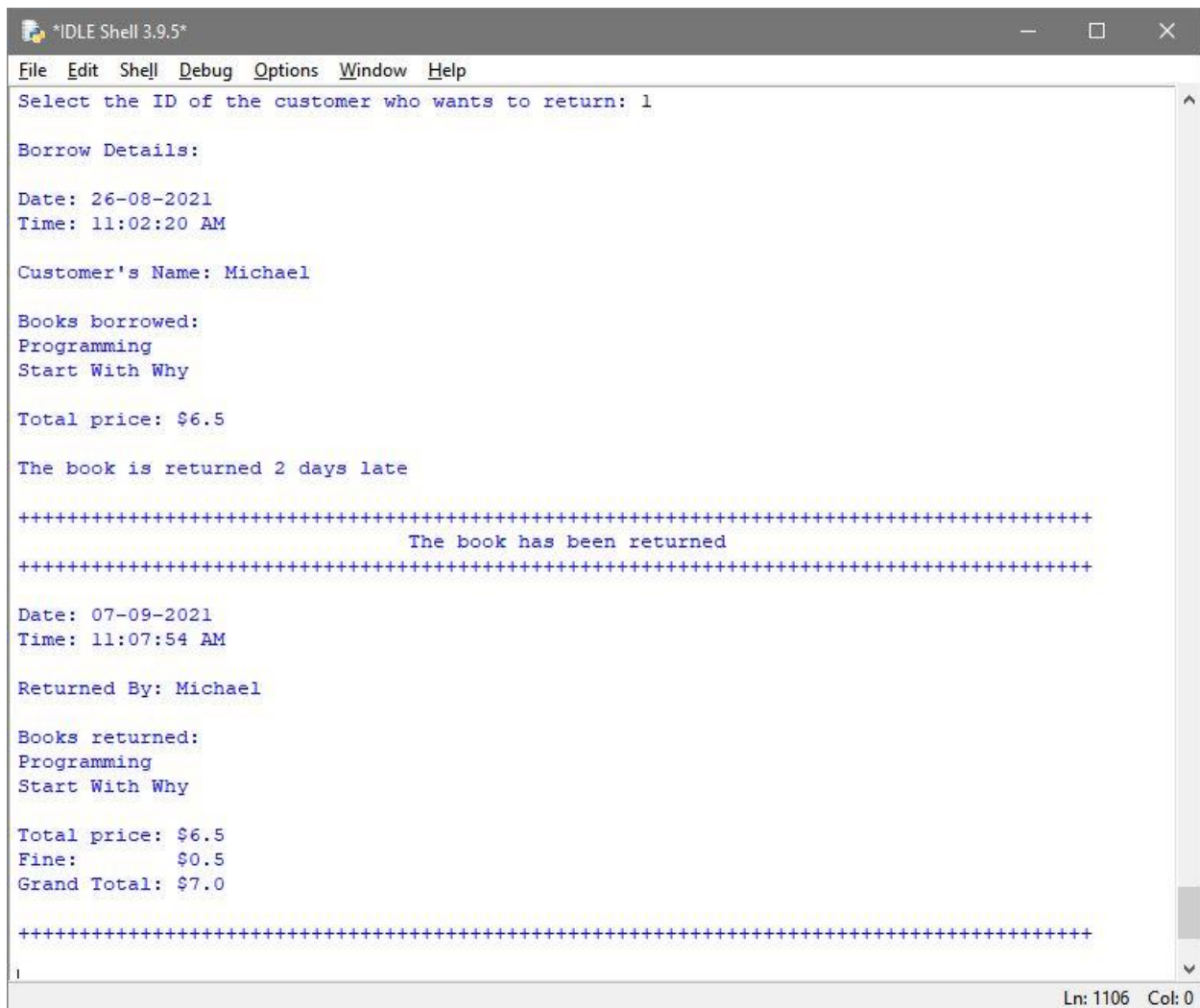
-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt
-----

Select the ID of the customer who wants to return: |

Ln: 115 Col: 51
```

Figure 35: Output when unspecified number is received as customer ID

If the ID of the customer is selected which matches the displayed ID, the borrow details of the customer is displayed. Also, if the customer returns the book after the lending period is over (i.e., after 10 days), a message is displayed showing the number of days which the book has been returned late. Then, it displays the return details where the fine is added by \$0.25/day is for the number of days returned late and the grand total amount which the customer has to pay is displayed.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Select the ID of the customer who wants to return: 1

Borrow Details:

Date: 26-08-2021
Time: 11:02:20 AM

Customer's Name: Michael

Books borrowed:
Programming
Start With Why

Total price: $6.5

The book is returned 2 days late

+++++
                        The book has been returned
+++++

Date: 07-09-2021
Time: 11:07:54 AM

Returned By: Michael

Books returned:
Programming
Start With Why

Total price: $6.5
Fine:      $0.5
Grand Total: $7.0

+++++
|
Ln: 1106 Col: 0
```

Figure 36: Output when appropriate customer ID is received

The bill generated for returning the books after the lending period is over is depicted below.

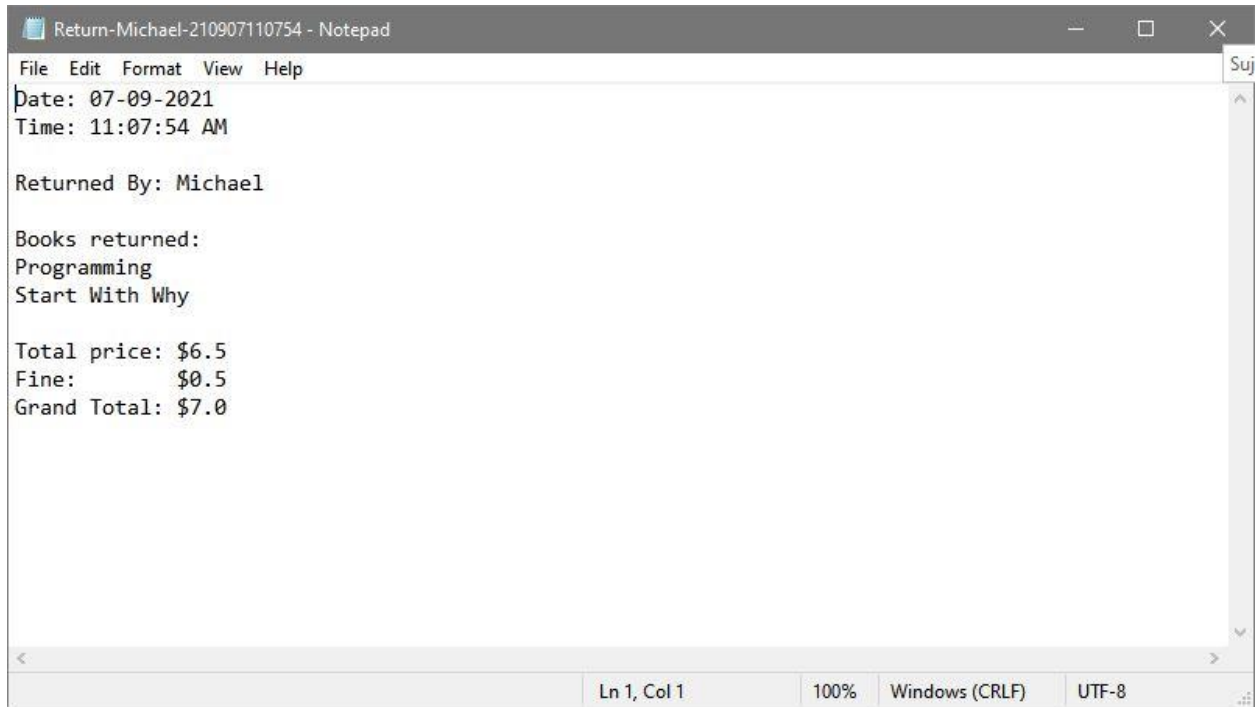
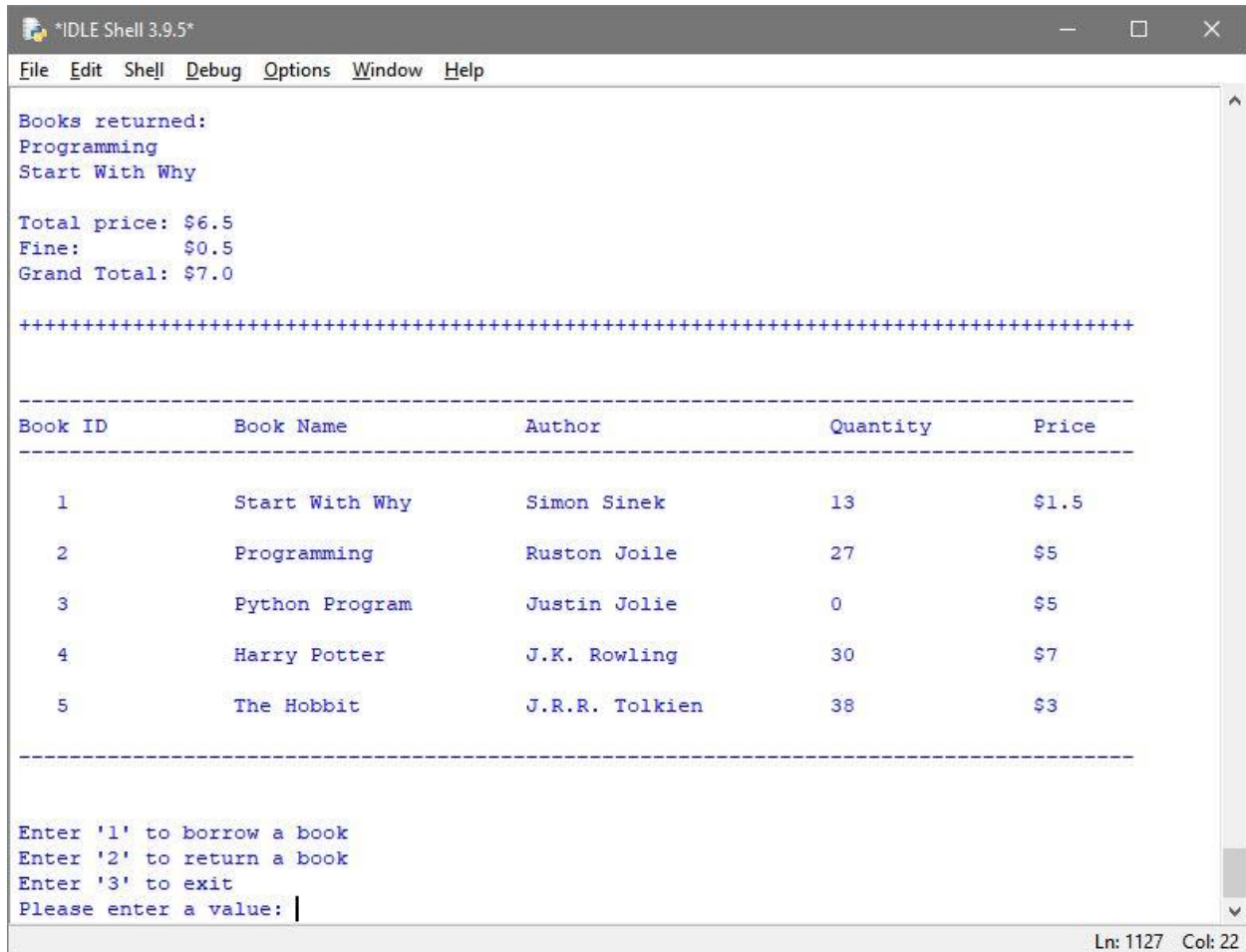


Figure 37: Bill generated after returning the books late

After that, the book details of the updated books are displayed and returns to the main screen and asks the user to input 1,2 or 3 to borrow the books, return the books or exit the program respectively.



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

Books returned:
Programming
Start With Why

Total price: $6.5
Fine:      $0.5
Grand Total: $7.0

+++++

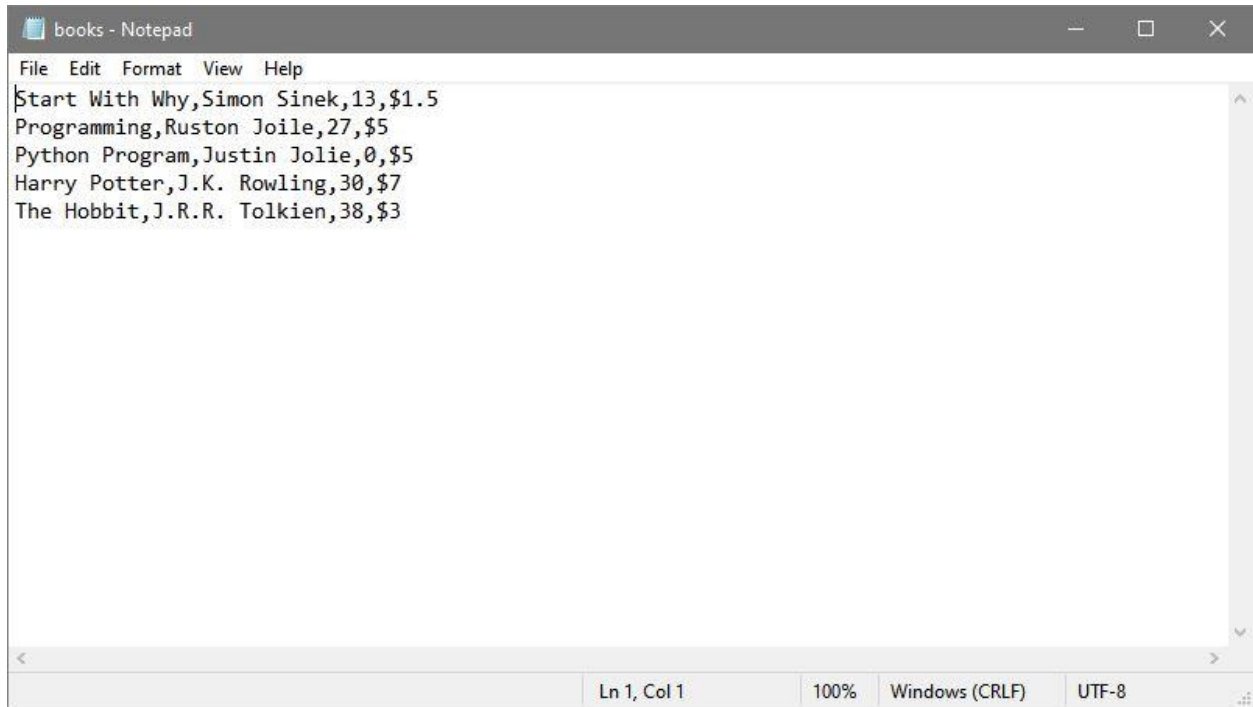
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    13           $1.5
2           Programming     Ruston Joile   27           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien 38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
```

Ln: 1127 Col: 22

Figure 38: Output when appropriate customer ID is received continued

The updated details in the stock file after the books have been returned is depicted below.



```
books - Notepad
File Edit Format View Help
Start With Why,Simon Sinek,13,$1.5
Programming,Ruston Joile,27,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,30,$7
The Hobbit,J.R.R. Tolkien,38,$3
Ln 1, Col 1 100% Windows (CRLF) UTF-8
```

Figure 39: Stock file after all books are returned

If 3 is entered as value, it displays an exit message and terminates the program.



```
IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    13           $1.5
2           Programming     Ruston Joile   27           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit       J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 3

+++++
                Thank You for using our Library Management System
+++++
>>> |
```

Ln: 1411 Col: 4

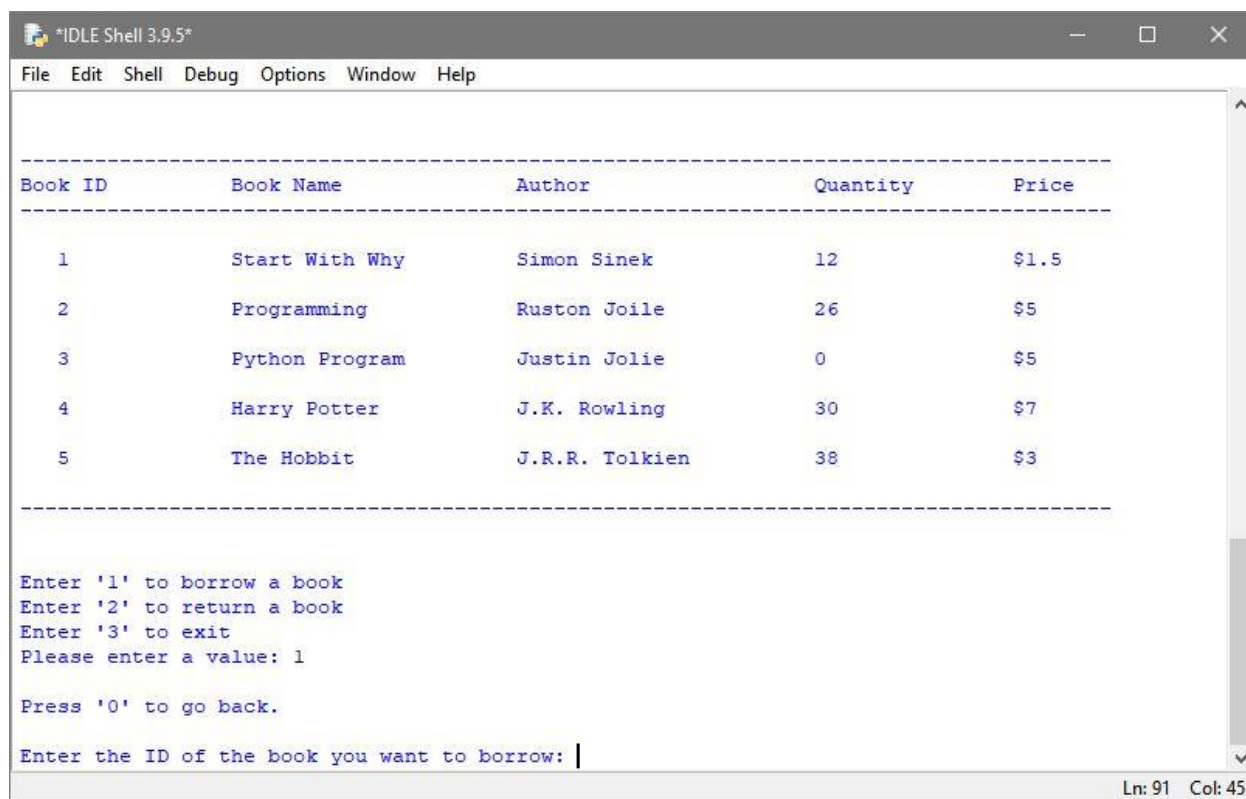
Figure 40: Output when 3 is received as value

4. Testing

Test 1

Objective	To show the implementation of try, except by displaying an error message when an unspecified value is entered.
Action	First, the program was opened and correct value was entered. Then, values other than specified values were entered.
Expected Result	The program should continue when valid input is entered and show an error message when unspecified value is entered.
Actual Result	The program continued when valid input was entered and showed an error message when unspecified value was entered.
Test	The test was successful.

Table 1: To test the implementation of try and except



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek    12           $1.5
2           Programming      Ruston Joile    26           $5
3           Python Program   Justin Jolie    0            $5
4           Harry Potter     J.K. Rowling    30           $7
5           The Hobbit       J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 1

Press '0' to go back.

Enter the ID of the book you want to borrow: |
```

Ln: 91 Col: 45

Figure 41: Test – Program running as required

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: one

+++++
Invalid input
Please provide value as 1, 2 or 3.
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek    12           $1.5
2           Programming      Ruston Joile    26           $5
3           Python Program   Justin Jolie    0            $5
4           Harry Potter     J.K. Rowling    30           $7
5           The Hobbit       J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
  
```

Ln: 44 Col: 0

Figure 42: Test – Error displayed when sting input

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 5

+++++
Invalid input
Please provide value as 1, 2 or 3.
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
Ln: 71 Col: 7

```

Figure 43: Test – Error displayed when unspecified number input as value

Test 2

Objective	To show an error message when incorrect id or name of the customer is entered.
Action	First, the borrow option was selected and a negative value was entered as book ID. Then, the return option was selected and a non-existent customer name was entered.
Expected Result	Error messages with suitable description should be displayed for both cases.
Actual Result	Error messages with suitable description were displayed for both cases.
Test	The test was successful.

Table 2: To test the result when incorrect id or name is entered

```

IDLE Shell 3.9.5
File Edit Shell Debug Options Window Help
Press '0' to go back.
Enter the ID of the book you want to borrow: one

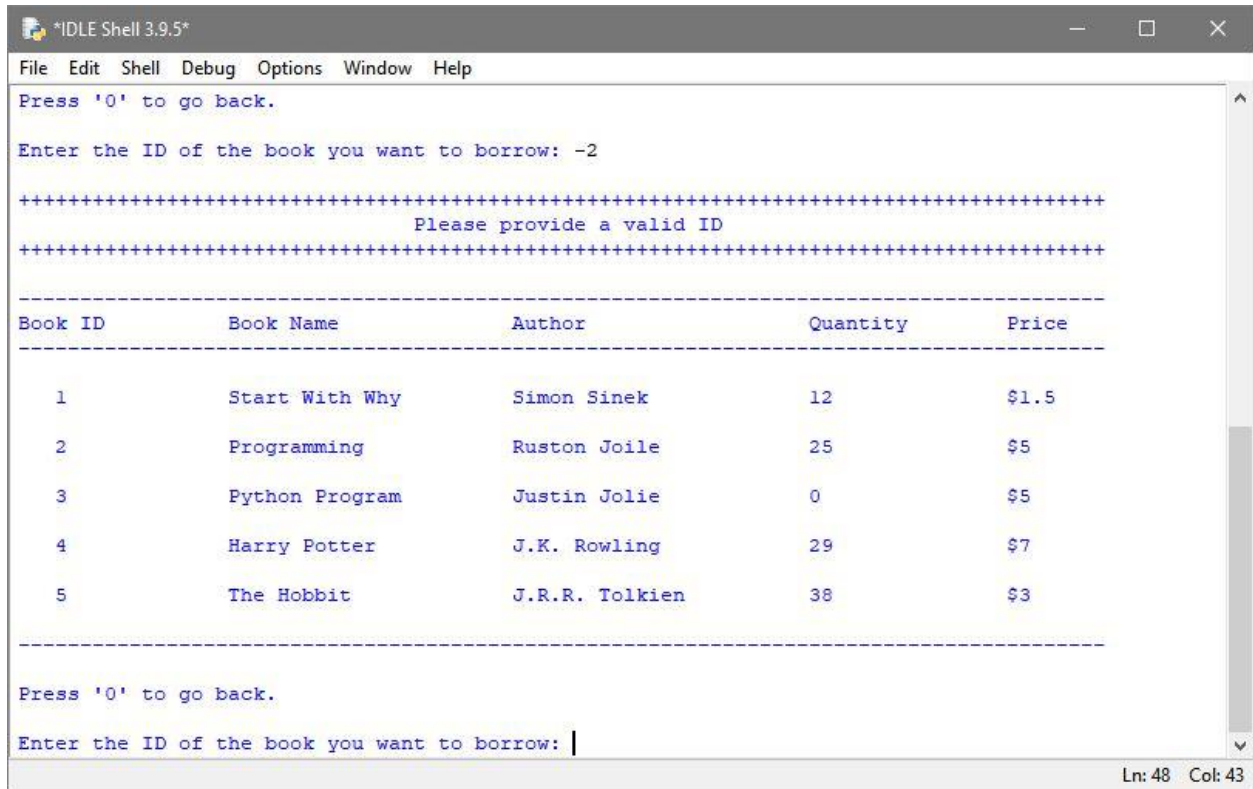
+++++
                Please provide a valid ID
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why   Simon Sinek    12           $1.5
2           Programming       Ruston Joile   26           $5
3           Python Program    Justin Jolie   0            $5
4           Harry Potter       J.K. Rowling   30           $7
5           The Hobbit         J.R.R. Tolkien 38           $3
-----

Press '0' to go back.
Enter the ID of the book you want to borrow: |
Ln: 289 Col: 0

```

Figure 44: Test – Error message displayed when string input as book ID



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Press '0' to go back.
Enter the ID of the book you want to borrow: -2

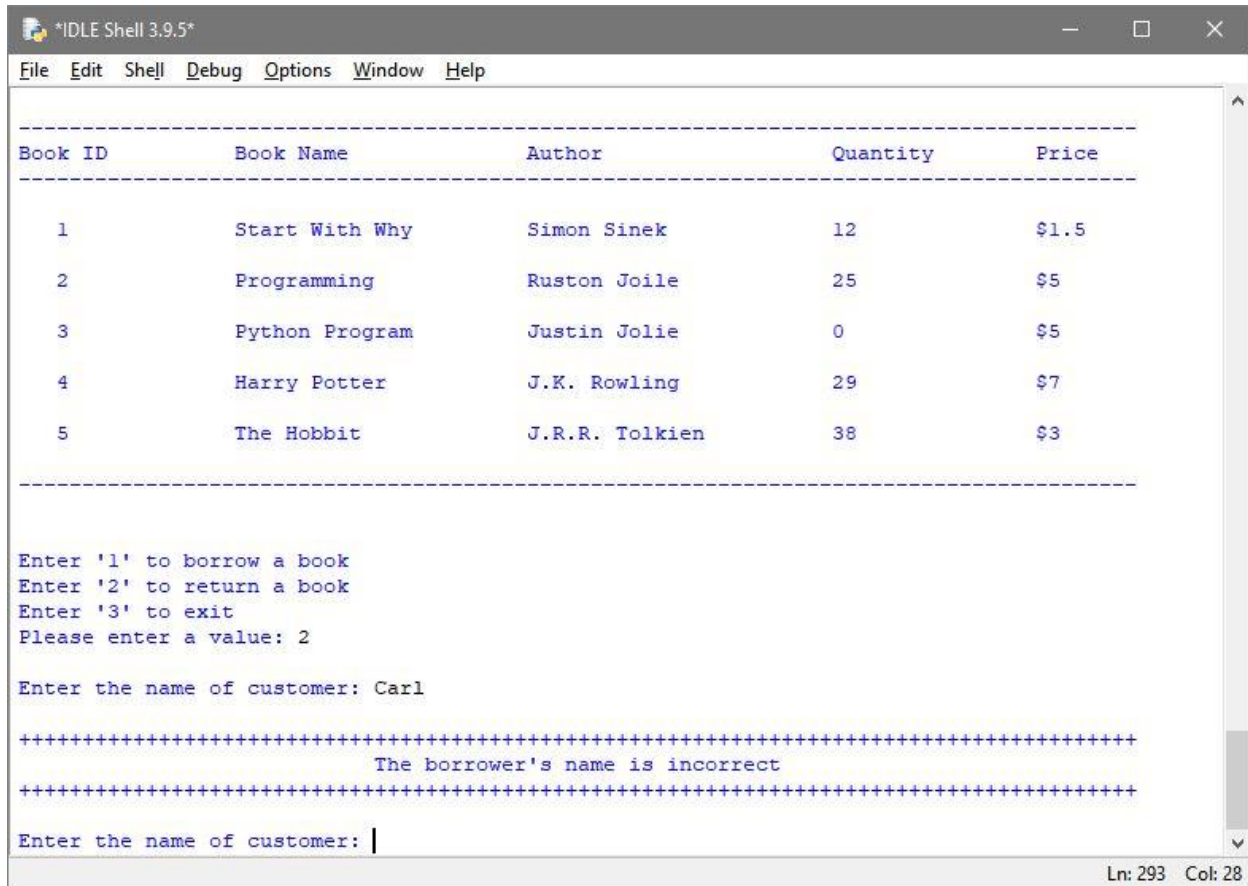
+++++
                        Please provide a valid ID
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why   Simon Sinek    12           $1.5
2           Programming         Ruston Joile   25           $5
3           Python Program      Justin Jolie   0            $5
4           Harry Potter         J.K. Rowling   29           $7
5           The Hobbit           J.R.R. Tolkien 38           $3
-----

Press '0' to go back.
Enter the ID of the book you want to borrow: |
```

Ln: 48 Col: 43

Figure 45: Test – Error displayed when unspecified number input as book ID



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   25           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: Carl

+++++
                        The borrower's name is incorrect
+++++

Enter the name of customer: |

Ln: 293 Col: 28
```

Figure 46: Test – Error displayed when incorrect borrower name entered

Test 3

Objective	To show complete process of borrowing the books.
Action	The borrow option was selected and the process was carried out completely.
Expected Result	The required output should be displayed in the shell as well as a note should be generated containing details of borrow.
Actual Result	The required output was displayed in the shell as well as a note was generated containing details of borrow.
Test	The test was successful.

Table 3: To test the complete process of borrowing the books

The screenshot shows a Python IDLE Shell window titled '*IDLE Shell 3.9.5*'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main text area displays a table of books with columns: Book ID, Book Name, Author, Quantity, and Price. The table lists five books: 1 (Start With Why, Simon Sinek, 12, \$1.5), 2 (Programming, Ruston Joile, 26, \$5), 3 (Python Program, Justin Jolie, 0, \$5), 4 (Harry Potter, J.K. Rowling, 30, \$7), and 5 (The Hobbit, J.R.R. Tolkien, 38, \$3). Below the table, the program prompts the user to enter a value to borrow, return, or exit a book. The user has entered '1', and the program prompts for the book ID to borrow. The status bar at the bottom right shows 'Ln: 91 Col: 45'.

```

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie    0           $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 1

Press '0' to go back.

Enter the ID of the book you want to borrow: |

```

Figure 47: Test - Output after receiving 1 as value

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

+++++
                        Book is available
+++++

Enter the name of Borrower: John
The date of borrow is: 01-09-2021
The time of borrow is: 10:37:14 AM
The price of the book is: $7.0

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why    Simon Sinek    12           $1.5
2           Programming      Ruston Joile   26           $5
3           Python Program   Justin Jolie   0            $5
4           Harry Potter     J.K. Rowling   29           $7
5           The Hobbit       J.R.R. Tolkien 38           $3
-----

Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: |
Ln: 74 Col: 0

```

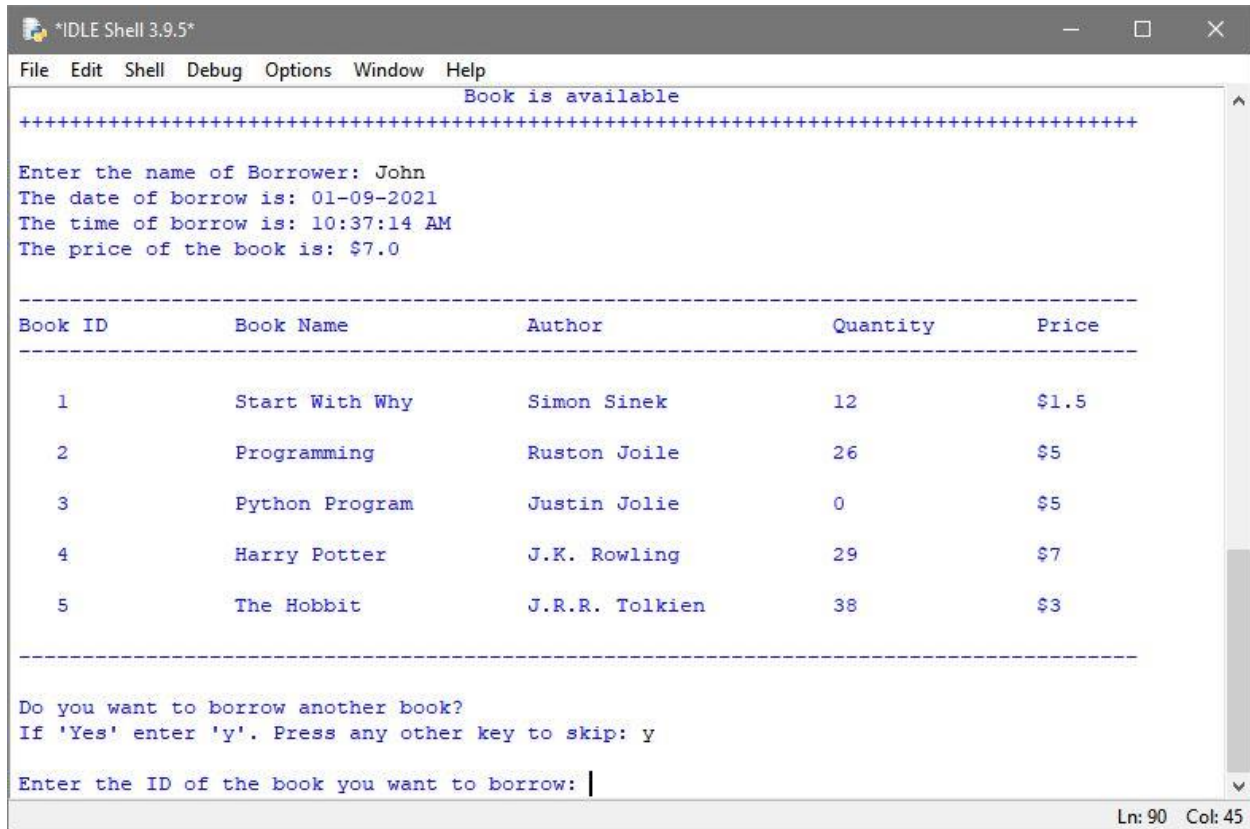
Figure 48: Test - Output after receiving borrower's name

```

books - Notepad
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,26,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,29,$7
The Hobbit,J.R.R. Tolkien,38,$3
|
Ln 6, Col 1  100%  Windows (CRLF)  UTF-8

```

Figure 49: Test - Stock file after a book is borrowed



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Book is available
+++++
Enter the name of Borrower: John
The date of borrow is: 01-09-2021
The time of borrow is: 10:37:14 AM
The price of the book is: $7.0

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming      Ruston Joile   26           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien 38           $3

-----
Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: y
Enter the ID of the book you want to borrow: |
```

Ln: 90 Col: 45

Figure 50: Test - Output after receiving 'y' as input to borrow another book

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
If 'Yes' enter 'y'. Press any other key to skip: y

Enter the ID of the book you want to borrow: 2

+++++
                        Book is available
+++++

The price of the book is: $5.0

-----
Book ID      Book Name      Author      Quantity    Price
-----
1           Start With Why    Simon Sinek    12          $1.5
2           Programming          Ruston Joile   25          $5
3           Python Program        Justin Jolie   0           $5
4           Harry Potter          J.K. Rowling  29          $7
5           The Hobbit             J.R.R. Tolkien 38          $3
-----

Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: |
Ln: 106 Col: 0

```

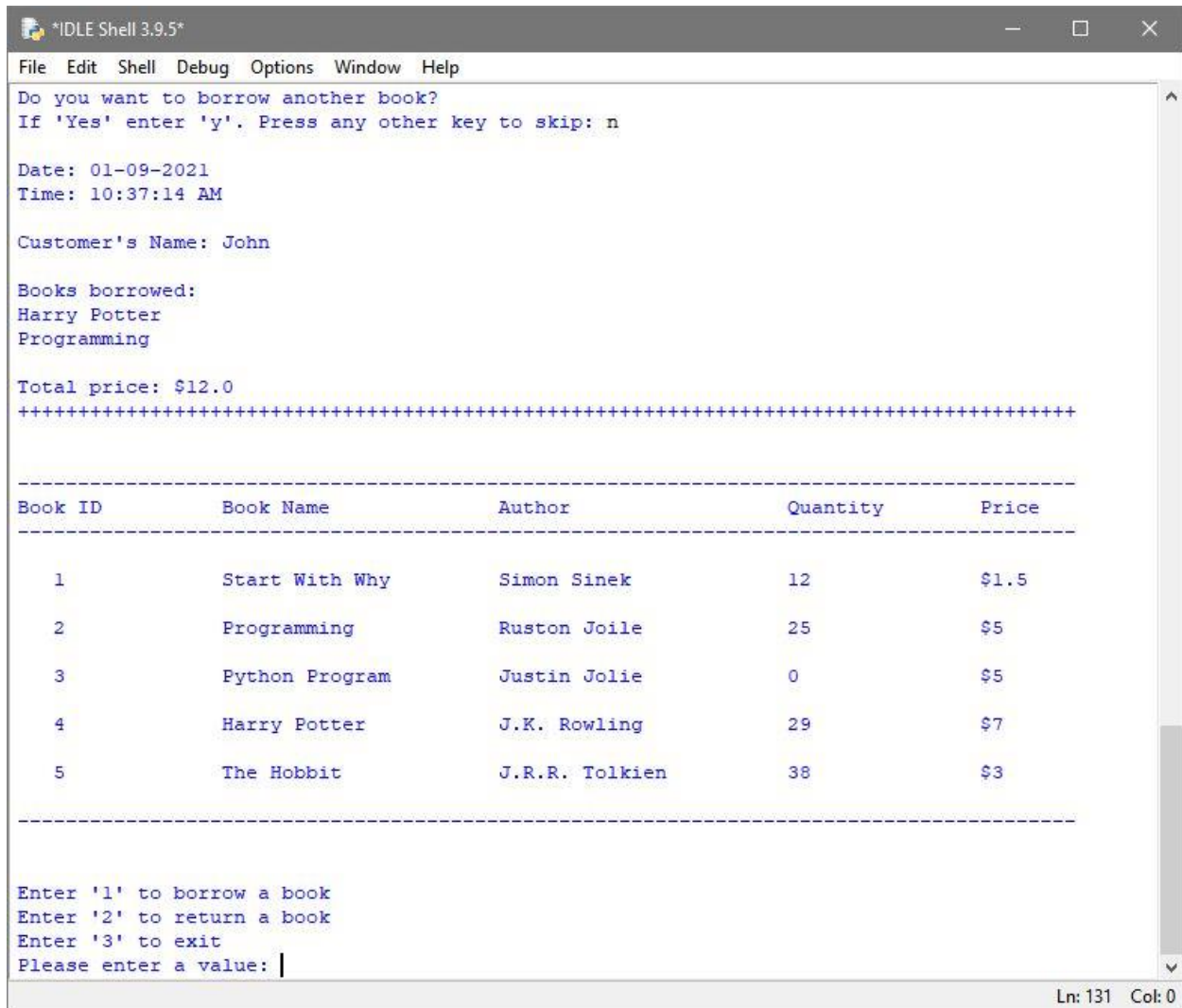
Figure 51: Test - Output after borrowing another book

```

books - Notepad
File Edit Format View Help
Start With Why,Simon Sinek,12,$1.5
Programming,Ruston Joile,25,$5
Python Program,Justin Jolie,0,$5
Harry Potter,J.K. Rowling,29,$7
The Hobbit,J.R.R. Tolkien,38,$3
Ln 2, Col 28  100%  Windows (CRLF)  UTF-8

```

Figure 52: Test - Stock file after another book is borrowed



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Do you want to borrow another book?
If 'Yes' enter 'y'. Press any other key to skip: n

Date: 01-09-2021
Time: 10:37:14 AM

Customer's Name: John

Books borrowed:
Harry Potter
Programming

Total price: $12.0
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek      12           $1.5
2           Programming     Ruston Joile     25           $5
3           Python Program  Justin Jolie     0            $5
4           Harry Potter    J.K. Rowling     29           $7
5           The Hobbit      J.R.R. Tolkien   38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
```

Ln: 131 Col: 0

Figure 53: Test - Output after borrow completion

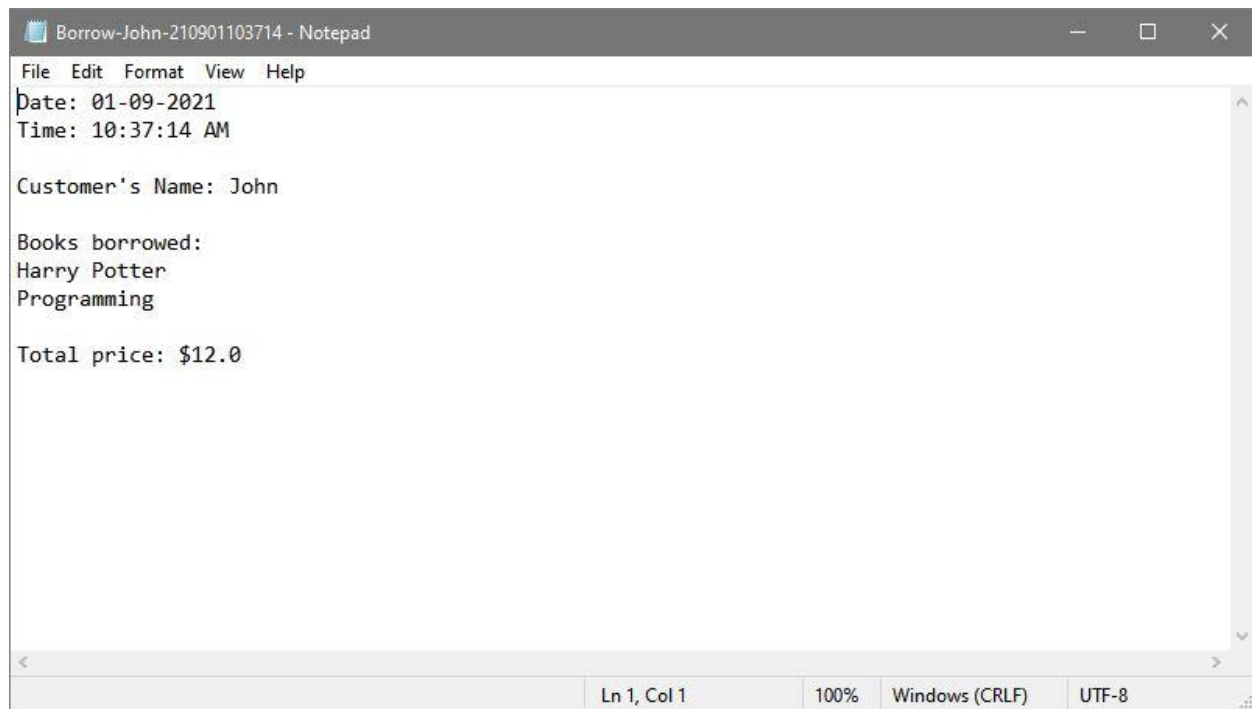


Figure 54: Test - Bill generated after borrowing the books

Test 4

Objective	To show the complete process of returning the books.
Action	The return option was selected and the process was carried out completely.
Expected Result	The required output should be displayed in the shell as well as a note should be generated containing details of borrow.
Actual Result	The required output was displayed in the shell as well as a note was generated containing details of borrow.
Test	The test was successful.

Table 4: To test the complete process of returning the books

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

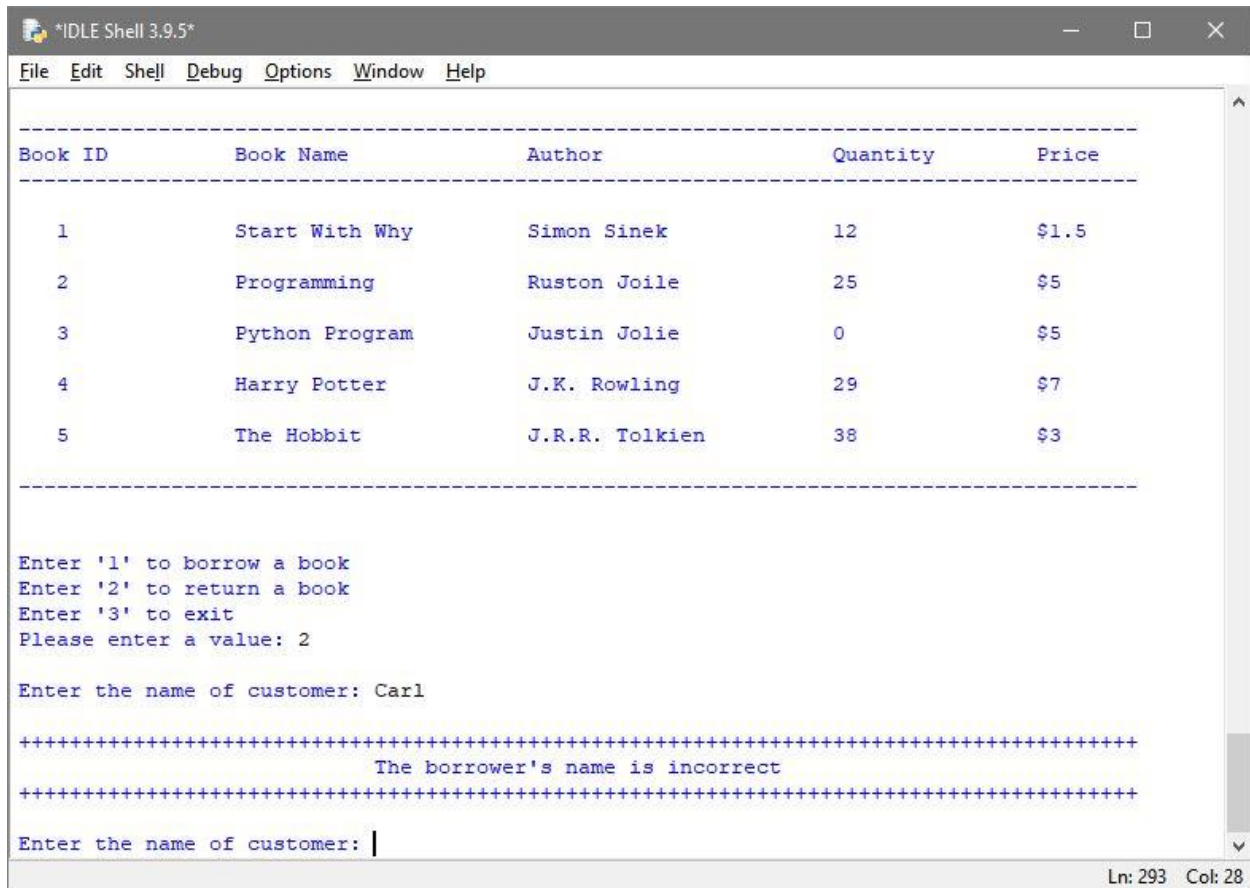
-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   25           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien 38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: |
Ln: 152 Col: 28

```

Figure 55: Test - Output after receiving 2 as input in value



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   25           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   29           $7
5           The Hobbit      J.R.R. Tolkien 38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: Carl

+++++
                        The borrower's name is incorrect
+++++

Enter the name of customer: |
```

Ln: 293 Col: 28

Figure 56: Test - Output after receiving unknown customer name



The screenshot shows a terminal window titled '*IDLE Shell 3.9.5*'. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The output text is as follows:

```
Enter the name of customer: John

Borrow Details:

Date: 01-09-2021
Time: 10:37:14 AM

Customer's Name: John

Books borrowed:
Harry Potter
Programming

Total price: $12.0

+++++
                        The book has been returned
+++++

Date: 05-09-2021
Time: 10:55:54 AM

Returned By: John

Books returned:
Harry Potter
Programming

Total price: $12.0
+++++
```

The status bar at the bottom right indicates 'Ln: 440 Col: 0'.

Figure 57: Test - Output after receiving valid customer name

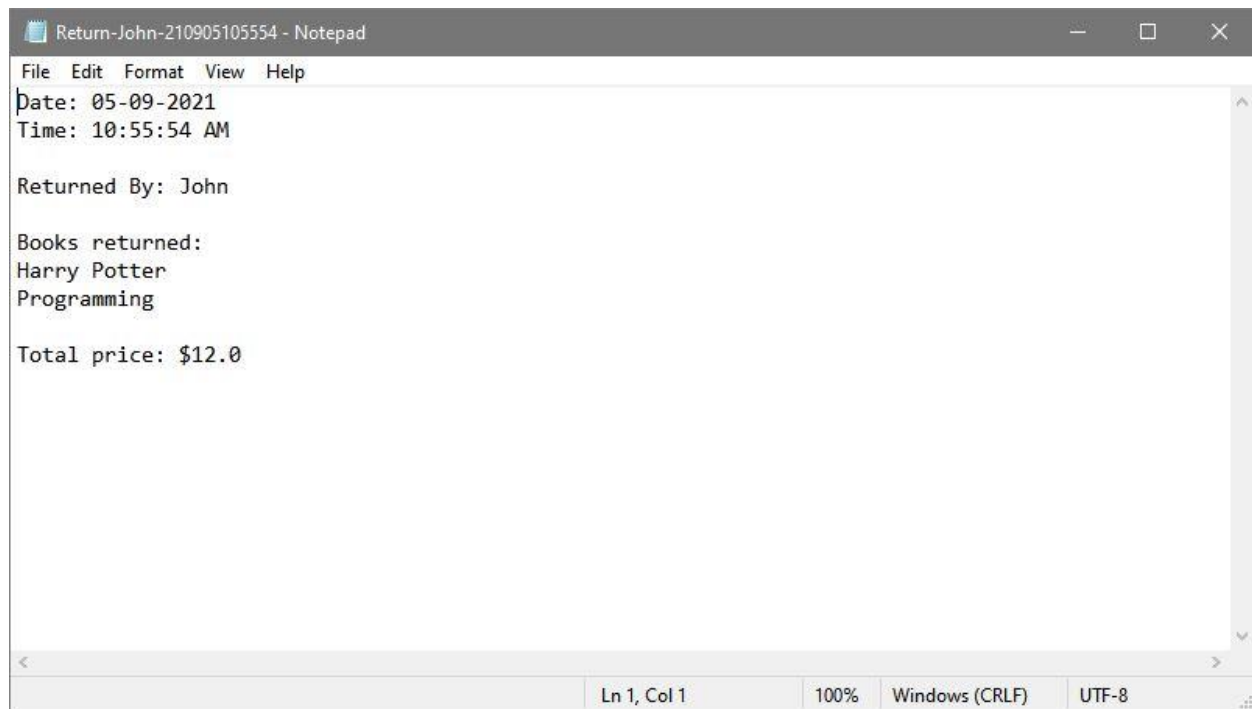
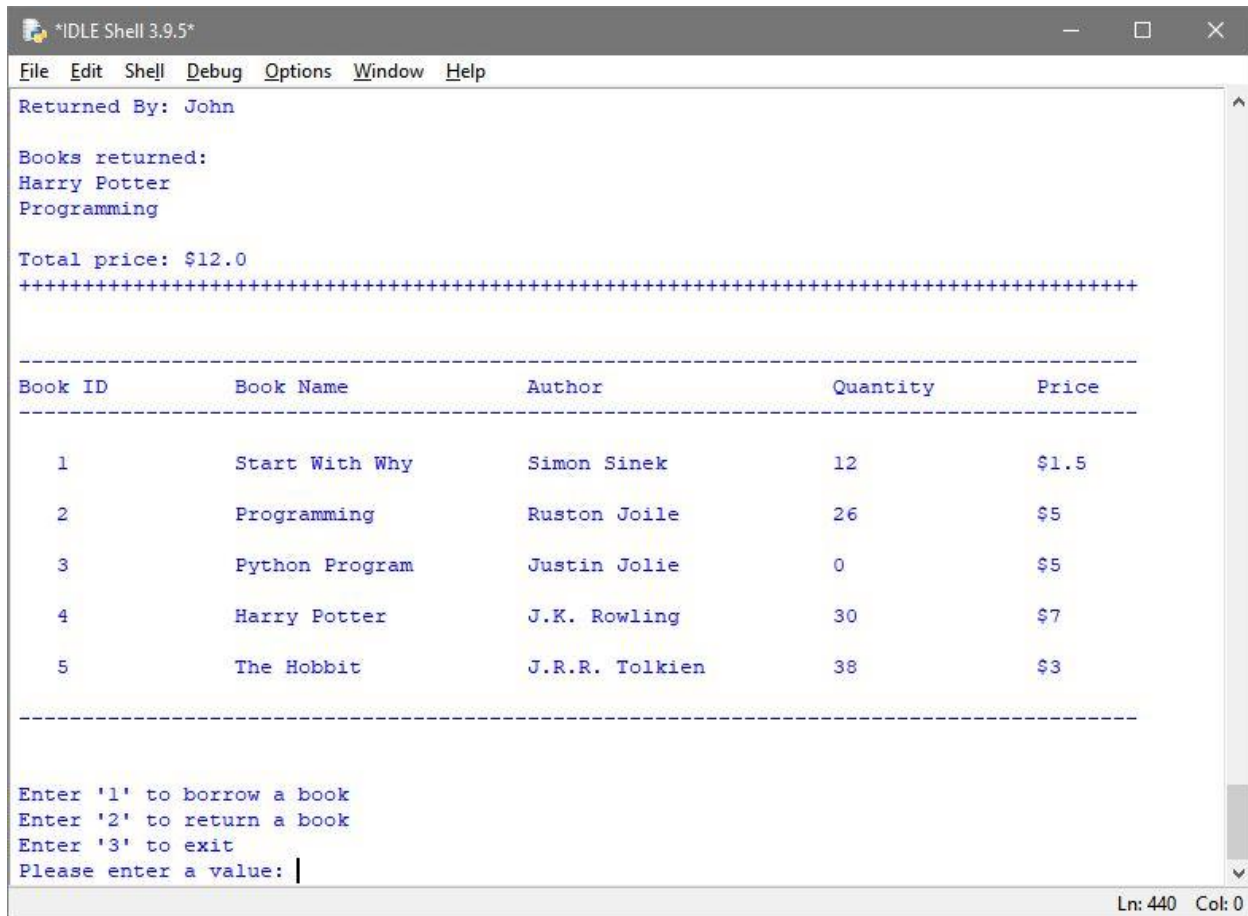


Figure 58: Test - Bill generated after returning the books



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Returned By: John

Books returned:
Harry Potter
Programming

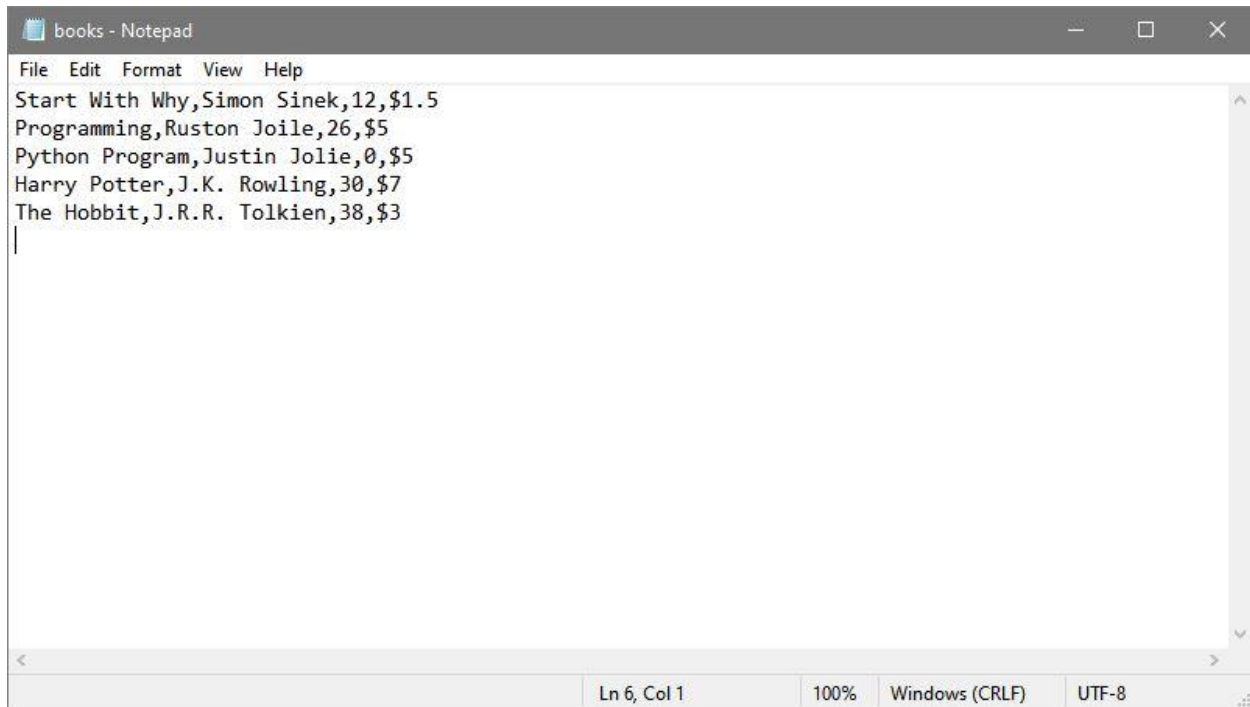
Total price: $12.0
+++++

-----
Book ID      Book Name      Author      Quantity      Price
-----
1           Start With Why  Simon Sinek    12           $1.5
2           Programming     Ruston Joile   26           $5
3           Python Program  Justin Jolie   0            $5
4           Harry Potter    J.K. Rowling   30           $7
5           The Hobbit      J.R.R. Tolkien  38           $3
-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: |
```

Ln: 440 Col: 0

Figure 59: Test - Output after receiving valid customer name continued



The image shows a Notepad window with the title 'books - Notepad'. The menu bar includes 'File', 'Edit', 'Format', 'View', and 'Help'. The text content is as follows:

```
Start With Why,Simon Sinek,12,$1.5  
Programming,Ruston Joile,26,$5  
Python Program,Justin Jolie,0,$5  
Harry Potter,J.K. Rowling,30,$7  
The Hobbit,J.R.R. Tolkien,38,$3  
|
```

The status bar at the bottom indicates 'Ln 6, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

Figure 60: Test - Stock file after returning the books

Test 5

Objective	To return the book after lending period (10 days) is over.
Action	The books were returned after the lending period was over.
Expected Result	A fine should be added to the total amount and bill should be generated accordingly.
Actual Result	A fine was added to the total amount and bill was generated accordingly.
Test	The test was successful.

Table 5: To test the return process after lending period is over

```

*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
1      Start With Why      Simon Sinek      12      $1.5
2      Programming      Ruston Joile      26      $5
3      Python Program      Justin Jolie      0      $5
4      Harry Potter      J.K. Rowling      30      $7
5      The Hobbit      J.R.R. Tolkien      38      $3

-----

Enter '1' to borrow a book
Enter '2' to return a book
Enter '3' to exit
Please enter a value: 2

Enter the name of customer: Michael

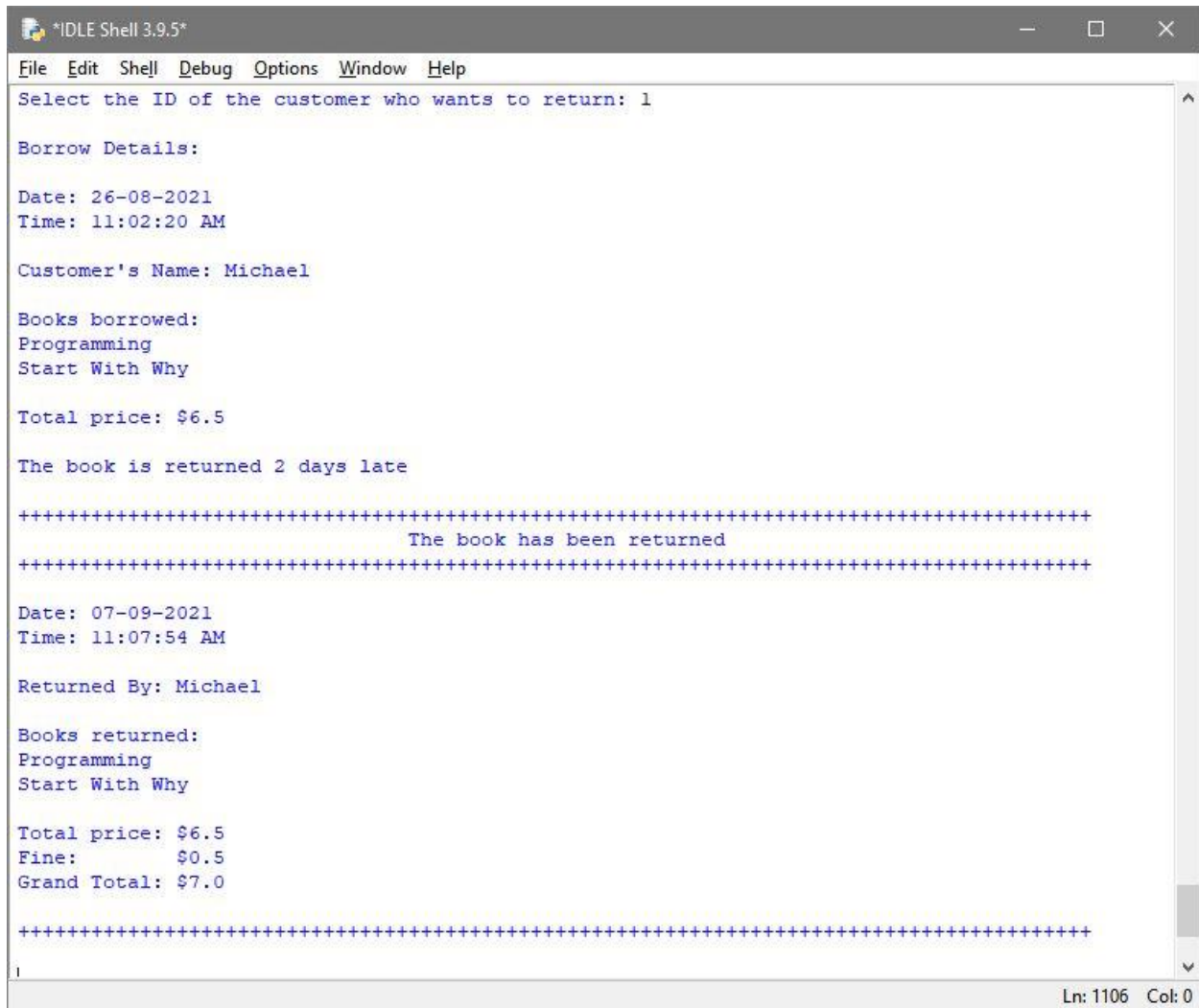
-----
Customer ID      Filename
-----
1      Borrow-Michael-210826110220.txt
2      Borrow-Michael-210830110556.txt

-----

Select the ID of the customer who wants to return: |
Ln: 1070 Col: 51

```

Figure 61: Test - Output when multiple customer name is same



```
*IDLE Shell 3.9.5*
File Edit Shell Debug Options Window Help
Select the ID of the customer who wants to return: 1

Borrow Details:

Date: 26-08-2021
Time: 11:02:20 AM

Customer's Name: Michael

Books borrowed:
Programming
Start With Why

Total price: $6.5

The book is returned 2 days late

+++++
                        The book has been returned
+++++

Date: 07-09-2021
Time: 11:07:54 AM

Returned By: Michael

Books returned:
Programming
Start With Why

Total price: $6.5
Fine:      $0.5
Grand Total: $7.0

+++++
|
Ln: 1106 Col: 0
```

Figure 62: Test - Output when appropriate customer ID is received

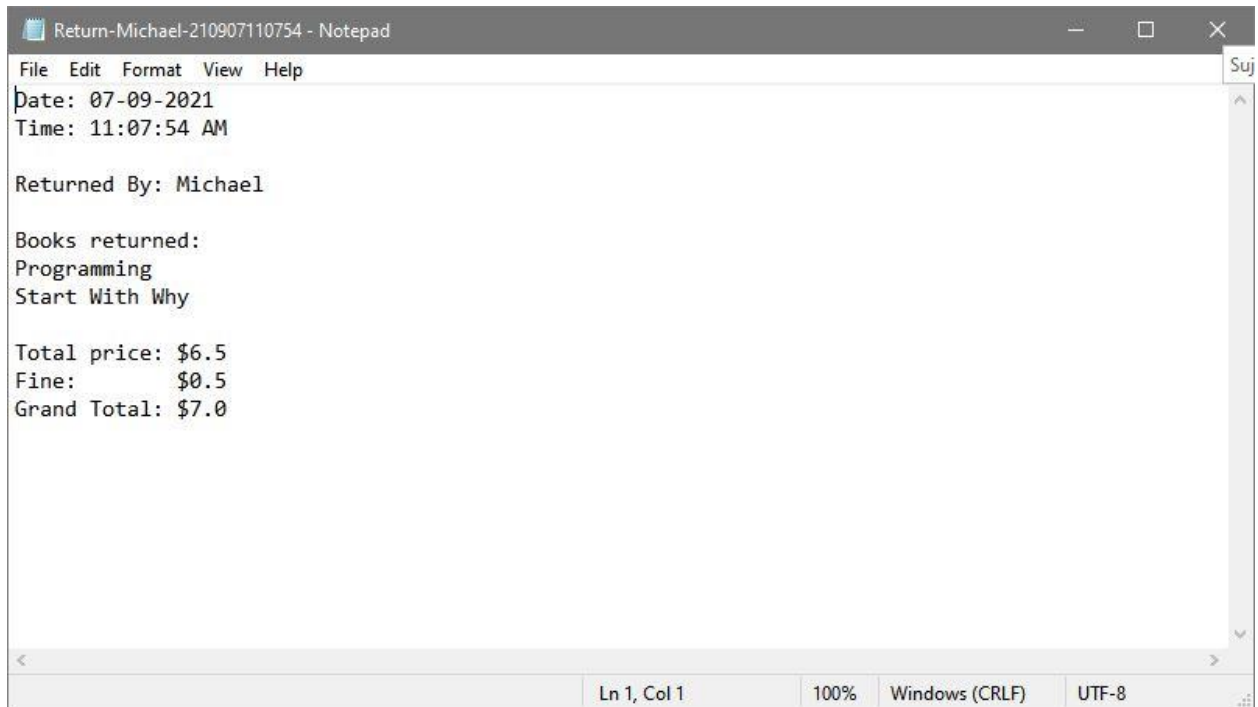


Figure 63: Test - Bill generated after returning the books late

5. Conclusion

In this project, I created a library management system using various tools like python, idle, notepad, draw.io and MS-Word. While working on this project, I learned about the various python functions, keywords, naming conventions and the methods to use them effectively. I learnt that creating a program helps to make the repetitive tasks execute quickly and efficiently and solves the recurrent problems. I understood that conditional and control flow statements are very important to direct the program and perform tasks as required. I also learnt to create the program in a modular way. This helped to minimize the repetitive codes and make the program more efficient.

I was not familiar with most of the terms related to the content. So, I searched about it in various articles online and looked into my lecture content. There were some specific ideas and concepts which I couldn't grasp while studying it on my own. So, I asked my lecturers to help me understand those topics. They were very helpful and supportive, and explained those concepts to me in a very simple way. This helped me to learn about those ideas easily and further try new things. I tried to implement the things taught to me in this coursework and some of them worked effortlessly while some of them did not execute the way I intended. So, I did some more research and corrected the issues to make them work as required.

After working on this project, I have learnt to create a program with all the necessary components. I learnt that before actually starting to write all the codes, developing an algorithm and flowchart helps a lot to create the structure of the program. Creating the algorithm and flowchart first, has helped me save a lot of time and effort to draft the idea before implementing it. It minimized the confusion while developing the actual website.

6. Appendix

main.py

```
import messages
import functions
import book_borrow
import book_return

def selection_message():
    """Displays the number of actions that can be performed in the program"""
    continueLoop = True
    while continueLoop == True:
        try:
            book_borrow.display()
            print()
            print("Enter '1' to borrow a book")
            print("Enter '2' to return a book")
            print("Enter '3' to exit")
            value = int(input("Please enter a value: "))
            print()

            if value == 1:
                book_borrow.borrow_book()

            elif value == 2:
                book_return.return_book()

            elif value == 3:
                continueLoop = False
                messages.exit_library()
```

```
        else:
            messages.invalid_input()

    except:
        messages.invalid_input()

messages.title()
selection_message()
```

book_borrow.py

```
import functions
import messages

def books_dictionary():
    """Reads text file contents and puts each line in a list as values of a dictionary
    and assigns auto increasing numbers as keys for each list"""
    file = open("books.txt", "r")
    booksInDictionary = {}
    bookID = 0
    for line in file:
        line = line.replace("\n", "")
        bookID += 1
        booksInDictionary[bookID] = line.split(",")
    file.close()
    return booksInDictionary

booksDictionary = books_dictionary()

def display():
    """Displays book details in a tabular form"""
```

```
print()
messages.minus()
print("Book ID" " \t Book Name" "\t\tAuthor" "\t\t\tQuantity" "\tPrice")
messages.minus()
print()
```

```
for key, value in booksDictionary.items():
    value = "\t\t".join(value) #Separates each list value in dictionary with 2 tab space
    print(" ", key, "\t\t", value)
    print()
messages.minus()
print()
```

```
def borrow_book():
    """Takes Book ID as input from user and gives suitable output according to the input
    provided
    and writes the details of borrowed books in a text file"""
    borrowLoop = True
    total = 0
    books = []
    while borrowLoop == True:
        try:
            if total == 0:
                print("Press '0' to go back.")
                print()
            b = int(input("Enter the ID of the book you want to borrow: "))
            for key,value in booksDictionary.items():
                if b == key:
                    book = booksDictionary[b][0]
                    quantity = int(booksDictionary[b][2])
                    price = float(booksDictionary[b][3].replace("$",""))
```

```

if quantity > 0:
    messages.available()
    books.append(book)
    if len(books) == 1:
        borrower = input("Enter the name of Borrower: ")
        print("The date of borrow is:", functions.date)
        print("The time of borrow is:", functions.time)
    if len(books) >= 1:
        print("The price of the book is: " + "$" + str(price))
        total = total + price

#Updates the book's quantity after a book is borrowed
file = open("books.txt","w")
for values in booksDictionary.values():
    #Matches the quantity of user input Book ID with quantity value in
dictionary;

    #Reduces the quantity in dictionary by 1
    if quantity == int(values[2]):
        values[2] = int(values[2]) - 1
        values[2] = str(values[2])
        file.write(str(values[0]) + "," + str(values[1]) + "," + str(values[2]) + ","
+ str(values[3]) + "\n")
    file.close()
display()
print("Do you want to borrow another book?")
answer = input("If 'Yes' enter 'y'. Press any other key to skip: ").lower()
print()
if answer != "y":
    borrowLoop = False
    customer = "Borrow-" + borrower + "-" + str(functions.unique) +
".txt"

```

```
#Writes customer details in a text file
file = open(customer,"w")
file.write("Date: " + functions.date + "\n")
file.write("Time: " + functions.time + "\n\n")
file.write("Customer's Name: " + borrower + "\n\n")
file.write("Books borrowed: " + "\n")
for i in range(len(books)):
    file.write(books[i] + "\n")
file.write("\nTotal price: " + "$" + str(total) + "\n")
file.close()
```

```
#Displays customer borrow details in shell
file = open(customer,"r")
for line in file:
    line = line.replace("\n","")
    print(line)
messages.plus()
file.close()
print()
```

```
else:
    messages.not_available()
```

```
elif b == 0:
    print()
    borrowLoop = False
    break
```

```
elif b > len(booksDictionary) or b < 0:
    messages.provide_valid_id()
```

```
        display()
        break

except:
    messages.provide_valid_id()
    display()
```

book_return.py

```
import messages
import functions
import book_borrow
import glob

def return_book():
    """Takes the borrower's name as input from user and returns the total amount to be
    paid as well as updates the stock of books"""
    returnLoop = True
    while returnLoop == True:
        try:
            name = input("Enter the name of customer: ")

            #Searches for the file names that start with the given parameters
            customer = (glob.glob("Borrow-" + name + ".*"))
            if len(customer) == 1:
                returner = customer[0]
            elif len(customer) > 1:
                idLoop = True
                while idLoop == True:
                    try:
                        print()
```

```
messages.minus()
print("Customer ID\t\t" + "Filename")
messages.minus()
print()

#Displays all the files having same customer name
for i in range (len(customer)):
    n = i + 1
    print("    " + str(n) + "\t\t" + customer[i])
    print()
messages.minus()
print()
c = int(input("Select the ID of the customer who wants to return: "))

if c > len(customer) or c < 1:
    messages.provide_valid_id()
else:
    r = c - 1 #Subtracts the input by 1 to match the index number
    returner = customer[r]
    idLoop = False
except:
    messages.provide_valid_id()

#Displays the previously borrowed information of the customer
books = [] #Stores all the borrow informaiton from text file of the customer
file = open(returner,"r")
print()
print("Borrow Details: ")
print()
lines = file.readlines()
for line in lines:
```

```
    line = line.replace("\n","")
    books.append(line)
    print(line)
print()
file.close()

returned = "Return-" + name + "-" + str(functions.unique) + ".txt"

#Creates a text file with the customer's information of the returned books
file = open(returned,"w")
file.write("Date: " + functions.date + "\n")
file.write("Time: " + functions.time + "\n\n")
file.write("Returned By: " + name + "\n\n")
file.write("Books returned: " + "\n")
for i in range(6, len(books)):
    file.write(books[i] + "\n")
from datetime import datetime
from datetime import timedelta
borrowDate = []
borrowDate.append(books[0].replace("Date: ",""))#Extracts time from borrow file

#Converts the date & time from string format to date & time format
startDate = datetime.strptime(borrowDate[0], "%d-%m-%Y")
dateToday = datetime.strptime(functions.date, "%d-%m-%Y")

endDate = startDate + timedelta(days=10)#Adds 10 days to borrow date
if dateToday > endDate:
    days = (dateToday - endDate).days
    fine = 0.25 * days
    total = float(books[9].replace("Total price: $",""))
    grand = fine + total
```



```

file.write("Fine:\t    " + "$" + str(fine) + "\n")
file.write("Grand Total: " + "$" + str(grand) + "\n\n")
print("The book is returned " + str(days) + " days late")
print()
file.close()

```

#Displays the customer information about the returned books

```

file = open(returned,"r")
messages.plus()
print("\t\t\t\t\tThe book has been returned")
messages.plus()
print()
for line in file:
    line = line.replace("\n","")
    print(line)
messages.plus()
file.close()
print()

```

#Updates the stock after books are returned

```

for i in range(6,len(books)-2):
    file = open("books.txt","w")
    for values in book_borrow.booksDictionary.values():
        if books[i] == values[0]:
            values[2] = int(values[2]) + 1
            values[2] = str(values[2])
            file.write(str(values[0]) + "," + str(values[1]) + "," + str(values[2]) + "," +
str(values[3]) + "\n")
    file.close()
returnLoop = False

```

```

except:
    print()
    messages.plus()
    print("\t\t\t The borrower's name is incorrect")
    messages.plus()
    print()

```

functions.py

```

def date_time():
    """Returns current date and time as a string. Also combines the date and time to give
    unique file name"""
    from datetime import datetime
    date = datetime.now().strftime('%d-%m-%Y')
    time = datetime.now().strftime('%I:%M:%S %p')
    unique = datetime.now().strftime('%y%m%d%I%M%S')
    return date, time, unique

#Declaring as global variables
date, time, unique = date_time()

```

messages.py

```

import book_borrow

def title():
    """Displays welcome message"""

    print("+++++
    +++++")
    print("\t\t\t Welcome to Library Management System")

```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
++++++++++++++++++++++++++++++++++++")  
    print()  
  
def available():  
    print()  
  
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
++++++++++++++++++++++++++++++++++++")  
    print("\t\t\t\t Book is available")  
  
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
++++++++++++++++++++++++++++++++++++")  
    print()  
  
def not_available():  
    print()  
  
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
++++++++++++++++++++++++++++++++++++")  
    print("\t\t\t\t Book is not available")  
  
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++  
++++++++++++++++++++++++++++++++++++")  
    print()  
    book_borrow.display()  
    print()  
  
def provide_valid_id():  
    print()
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
    print("\t\t\tPlease provide a valid ID")
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
def exit_library():
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
    print("\t\t Thank You for using our Library Management System")
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
def invalid_input():
```

```
    print()
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
    print("\t\t\t Invalid input")
```

```
    print("\t\t\t Please provide value as 1, 2 or 3.")
```

```
print("+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++")
```

```
    print()
```

```
def plus():
```

```
print("++++  
++++")
```

```
def minus():
```

```
    print("-----")
```

Bibliography

Computer Hope, 2021. *What is a Program?*. [Online]

Available at: <https://www.computerhope.com/jargon/p/program.htm>

[Accessed 06 September 2021].

Hebb, N., 2021. *What is a Flow Chart ?*. [Online]

Available at: <https://www.breezetre.com/articles/what-is-a-flow-chart>

[Accessed 09 September 2021].

Jaiswal, S., 2017. *Python Data Structures Tutorial*. [Online]

Available at: <https://www.datacamp.com/community/tutorials/data-structures-python#files>

[Accessed 09 September 2021].

TechTarget Contributor, 2021. *What is algorithm?*. [Online]

Available at: <https://whatis.techtarget.com/definition/algorithm>

[Accessed 09 September 2021].

The Economic Times, 2021. *Software-Development*. [Online]

Available at: <https://economictimes.indiatimes.com/definition/pseudocode>

[Accessed 09 September 2021].