**IT 114 - ADVANCED PROGRAMMING FOR INFORMATION TECHNOLOGY**

**Case Studies**

**1. Formulating the Problem**

**1.1 Problem Description**

We must create a program that keeps track of books in a bookstore. If the book in the bookstore isn’t available, the program must ask the user if they would like to order it by filling out the book information. Information about the Book should include the Title, Author, ISBN, Genre and Price. Author’s can only have a maximum of 5 books.

The user should have the ability to search for authors, print a list of all authors and books in the system. The maximum number of authors in the system in 30, after which no more authors can be added.

**1.2 Verbalization**

*What is the goal?*

Create a program that keeps track of books and allows the user to add additional authors/books if its not shown.

*What are the givens?*   The variables of the books (ISBN, Title, Author, Genre, Price), how many books the author can have, how many authors can be in the system.

*What are the unknowns?*   The books in the system

1.3 **Information Elicitation**

*Goal*

Create a program that -- keeps track of books --- allows the user to add additional authors/books if not shown --- allow user to search for Authors --- allow user to show all authors and books.

*Givens*   The variables of the books, how many books the author can have, how many authors can be in the system.

*Unknowns*   What books and authors are in the system

*Conditions*   Authors cannot have more than 5 books in the system. There can be no more than 30 Authors in the system.

**2. Planning the Solution**

**2.1 Solution Strategy**

Create a library of books and authors for the user to see/choose from initially. Prompt the user to perform an action (be it a search or print). Add the author and book info if not in the system and author caps aren’t reached.

**)2.2 Goal Decomposition**

*Sub-goal 1*

Create an array of books/authors for the user to see.

*Sub-goal 2*

Limit the number of books an author can have

*Sub-goal 3*

Limit the number of authors in the system

*Sub-goal 4*

Allow the user to choose search the system for authors

*Sub-goal 5*

Add Author/book info to system if not in it.

*Sub-goal 6*

Allow user to see all authors/books

**2.3 Resources**

*Relevant Books*

**Title:** Catcher in the Rye; The Outsiders; The Giver; Harry Potter and the Sorcerer's Stone; Harry Potter and the Chamber of Secrets; Harry Potter and the Prisoner of Azkaban; Harry Potter and the Goblet of Fire; Harry Potter and the Order of the Phoenix;

**Authors:** Salinger, Hinton, Lowry, Rowling

**ISBN:** 11511151; 1212154; 13245616; 465498465; 4194119; 559121; 131881; 33652

**Price:** 4.93; 14.00; 6.99; 8.99; 8.99; 8.99; 8.99; 8.99

**Genre:** Literary Realism; Young Adult Fiction; Young Adult Fiction; Fantasy Fiction; Fantasy Fiction; Fantasy Fiction; Fantasy Fiction; Fantasy Fiction

**3. Designing the Solution**

**)3.1 Structure Chart**

*First Level Decomposition*



*Goal Refinement*

**Sub-goal 1**

Create an array of books/authors for the user to see.

**Sub-goal 2**

Limit the number of books an author can have to 5

* Create a Boolean expression that returns true or false when the number of books an author has is <= 5

**Sub-goal 3**

Limit the number of authors in the system

* Create a Boolean expression that returns a true or false value when the number of authors in the system is <= 30

**Sub-goal 4**

Allow the user to choose to search the system for authors

* User input will give them the ability to search the system for their Author of join, if it’s not in there continues with goal 5 function

**Sub-goal 5**

Add Author/book info to system if not in it.

* The system will ask for Last, First names then the book information including (Title, ISBN, Genre, Price)

**Sub-goal 6**

Allow user to see all authors/books

* User input will trigger the print function to show all the author’s and their books in the system

*Second Level Decomposition*



The second level decomposition shows operations between the User ans System. The process starts with the user being asked what they would like to do (Print, Search, Exit). Search Prompts the user to search for an Author to show their books, if author not found it asks the User to input Author and Book info to add to the system. After it asks if the user wants to add another. Yes, loops back to prompting for author info and book info. No, brings the user back to the selection. Upon Selecting Print, the system checks its library for Books and Authors, If nothing it ends. If there is it prints all Authors and Books. Upon selecting Exit, the program ends.

**3.2 Module and Data Specifications**

**Name**: Instruct - Display instructions to the user

**Input**: 1

**Output**: All Authors and their Books

**Logic**: Retrieve all the Author’s and their books from the array

**Name**: Instruct - Display instructions to the user

**Input**: 2

**Output**: Prompt to search for Author

**Logic**: Asks the User for Author Name

**Name**: Search for Author

**Input**: Last name and First name

**Output**: Author details or none

**Logic**: System pulls information matching author name and prints out author and book info. If nothing matches, then user is prompted to add it.

**Name**: Add Author and Book info

**Input**: Author Last, First name and Book Information

**Output**: none

**Logic**: Take the input from the name, store them in a temp variable, use the temp variable to add them to the author array, when the book info is taken, add that to the author from the temp variable.

**Name**: Instruct - Display instructions to the user to add more or no

**Input**: yes/no

**Output**: if yes repeat last step of adding book info, if no next prompt

**Logic**: while loop makes it so that as long as its yes, the system will keep adding more. If no the user is prompt with next step

**Name**: Instruct - Display instructions to the user

**Input**: 3

**Output**: system closes

**Logic**: the operation is ended

**3.3 Logic**

*Logic*

1.0 Display user instructions

2.0 Upon Selecting Show all Retrieve Information on all Author’s and their books

2.1 Print the information about Author’s and their books

3.0 Upon Search Retrieve Information on the author and their books

3.1 Search the Array for the corresponding variables of Last and First to do this

3.2 If Found print the information about the author and their books

3.3.1 If not found retrieve input from user about author and book information

3.3.2 store it in array

4.0 Upon Selecting Exit

4.1 Exit/Close operation

*Algorithm Description*

The process of doing this : Data is stored in an array. The system gives the User an option to Print, Search or Exit this is done by typing either option 1 1 2 or 3. If the User selects Print (1) the system searches for Authors and books in the system. If nothing is there it exits. If there is, it prints all Author’s and their respected books, including the book info.

Search (2) lets the User search for an Author which then prints out that Author’s books and book info. If author isn’t there it prompts the user to add them and add the book info. It then adds the info to the appropriate array. After it asks the user if they would like to add more or do something else. If they choose yes, it repeats the action of adding more authors/books. If no it prompts the user the same functions as in the menu.

When the user selects Exit (3), the system closes/ends

**4. Translation**

**4.1** **Source Code**

I removed the indents to try to make it easier to read in this document. With the indents it was hard to look at.

//===================================================   
// Name : Emad Tirmizi  
// SID : 31400222  
// Course : IT114   
// Section :   
// Instructor : Maura Deek  
// T.A :   
//===================================================   
//===================================================   
// Assignment # : 1  
// Date : 10/4/2018  
//===================================================   
//===================================================   
// Description: This program will create a library of  
// books and authors for the user to see/choose from   
// initially. Upon Running it will prompt the user to  
// perform an action (be it a search or print). Add   
// the author and book info if not in the system and   
// author caps aren’t reached.   
//===================================================   
  
  
**import** java.util.**\***;  
//create a book class  
**class** Book{  
**private** String Title;  
**private** String ISBN;  
**private** String Genre;  
**private** **double** Price;  
//create variables   
//create book class for the book's information, this defines what the book is

**public** Book(String Title, String ISBN, String Genre, **double** Price){  
**this**.Title **=** Title;  
**this**.ISBN **=** ISBN;   
**this**.Genre **=** Genre;  
**this**.Price **=** Price;  
//give them values  
}//end for Book method  
//set and get to retrieve and store the values for the variables to be printed in the toString when Book is called upon  
**public** **void** setTitle(String s){  
**this**.Title **=** s;  
}//end for setTitle  
**public** **void** setISBN(String s){  
**this**.ISBN **=** s;  
}//end for setISBN  
**public** **void** setGenre(String s){  
**this**.Genre **=** s;  
}//end for setGenre  
**public** **void** setPrice(**double** Price){  
**this**.Price **=** Price;  
}//end for setPrice  
**public** String getTitle(){  
**return** Title;  
}//end for getTitle  
**public** String getISBN(){

**return** ISBN;  
}//end for getISBN  
**public** String getGenre(){  
**return** Genre;  
}//end for getGenre  
**public** **double** getPrice(){  
**return** Price;  
}//end for getPrice  
**public** String toString(){  
**return** "Book Title: " **+** Title **+** "\n" **+** "ISBN: " **+** ISBN **+** "\n" **+** "Genre: " **+** Genre **+** "\n" **+** "Price: " **+** Price **+** "\n\n";  
}//end for toString  
//create string that will be printed when book is displayed  
}//end for Book Class  
//create class for Author's books  
**class** Author{  
**private** String Last;  
**private** String First;  
//create an array for books by the author  
**private** ArrayList**<**Book**>** book **=** **new** ArrayList**<**Book**>**();  
//create Author method to be used when assiging books to authors  
**public** Author(String Last, String First){  
**this**.Last **=** Last;  
**this**.First **=** First;  
}//end for Author method

**public** **void** setLast(String s){  
Last **=** s;  
}//end for setLast  
**public** **void** setFirst(String s){  
First **=** s;  
}//end for setFirst  
**public** **void** setBooks(ArrayList**<**Book**>**b){  
book **=** b;  
}//end for setBooks  
**public** String getLast(){  
**return** Last;

}//end for getLast  
**public** String getFirst(){  
**return** First;  
}//end for getFirst  
**public** ArrayList**<**Book**>** getBooks() {  
**return** book;  
}//end for getBooks  
//get and set the values for the variables of book, author last, first name  
//create a function for adding books into the array for the author  
**public** **boolean** addBook(Book b){  
**if**(book.size()**<=**5){  
book.add(b);  
**return** **true**;  
}//end for if  
**else**{  
**return** **false**;  
}//end for else  
}//end for addBook method  
**public** String toString(){  
**return** Last **+** ", " **+** First **+** "\n" **+** "Book: " **+** book;  
}//end for toString method  
}//end for Author class  
//create a class for an array that holds and add Authors to the array  
**class** AddAuth{  
**private** ArrayList**<**Author**>** Author **=** **new** ArrayList**<**Author**>**();  
**public** ArrayList**<**Author**>** getAuthor(){  
**return** Author;  
}//end for ArrayList metod  
//if array size is less than or equal to 30 the system will return true and the boolean command will let you add them to the list  
//if they are over 30 then the system will return false and the command will not be executed  
**public** **boolean** addAuthor(Author A){  
**if**(Author.size()**<=**30){  
Author.add(A);  
**return** **true**;  
}//end if  
**else**{  
**return** **false**;  
}//end else  
}//end addAuth method  
}//end addAuth class  
//create a main method to run the system  
**public** **class** Main{  
**public** **static** **void** main(String[] args){   
//create books to be added into the system so that its not empy.  
//The order displayed is Title, ISBN, Genre, Price  
Book b1 **=** **new** Book("Catcher in the Rye", "11511151", "Literary Realism", 4.93);   
Book b2 **=** **new** Book("The Outsiders", "1212154", "Young Adult Fiction", 14.00);   
Book b3 **=** **new** Book("The Giver", "13245616", "Young Adult Fiction", 6.99);   
//These are only sperated by spaced because it was making it easier for me to assign them to the correct authors.   
Book b4 **=** **new** Book("Harry Potter and the Sorcerer's Stone", "465498465", "Fantasy Fiction", 8.99);  
Book b5 **=** **new** Book("Harry Potter and the Chamber of Secrets", "4194119", "Young Adult Fiction", 8.99);  
Book b6 **=** **new** Book("Harry Potter and the Prisoner of Azkaban", "559121", "Young Adult Fiction", 8.99);  
Book b7 **=** **new** Book("Harry Potter and the Goblet of Fire", "131881", "Young Adult Fiction", 8.99);  
Book b8 **=** **new** Book("Harry Potter and the Order of the Phoenix", "33652", "Young Adult Fiction", 8.99);  
//add authors to the array  
//add a book to that author specifically

AddAuth addA **=** **new** AddAuth();  
Author A1 **=** **new** Author("Salinger", "Jerome");  
A1.addBook(b1);  
//Adding programmed authors  
addA.addAuthor(A1);   
//add authors to the array  
//add a book to that author specifically   
Author A2 **=** **new** Author("Hinton", "Susan");  
A2.addBook(b2);  
//Adding programmed authors  
addA.addAuthor(A2);  
//add authors to the array  
//add a book to that author specifically   
Author A3 **=** **new** Author("Lowry", "Lois");  
A3.addBook(b3);  
//Adding programmed authors  
addA.addAuthor(A3);   
//add authors to the array  
//add a book to that author specifically  
//I gave J.K Rowling more books to demonstrate it not being allowed to add more than 5 books.  
Author A4 **=** **new** Author("Rowling", "Joanne");  
//Add the Books to the author  
A4.addBook(b4);  
A4.addBook(b5);  
A4.addBook(b6);  
A4.addBook(b7);  
A4.addBook(b8);  
//Adding programmed authors  
addA.addAuthor(A4);  
//Create a selection screen for the user to interact with  
System.out.println(" Enter '1': to Show all authors \n Enter '2': to Search for an Author" **+** "\n Enter '3': to Exit");  
//This picks up the input from the user  
Scanner sc **=** **new** Scanner(System.in);  
//The int option is for the selection screen. I found it the easiest way to use an int value rather than a string for the choices   
**int** option **=** sc.nextInt();  
**while**(option**!=** 3) {  
sc.nextLine();   
//if 1 is entered into the system. The function calls Author to display all the authors and books stored.  
**if**(option **==** 1) {  
**for**(Author A **:** addA.getAuthor()) {  
System.out.println(A);  
}//end for  
}//end if  
//if 2 is selected the system asks to enter the Authors Name  
**else** **if**(option **==** 2) {  
//The point of the check is to see if the Author is or isn’t in the library. If it is the check will = 1, if not check will = 0.  
**int** check **=** 0;

String selection;  
System.out.print("Please Enter Author's Last Name: ");  
String Last **=** sc.nextLine();

//Value gets stored in Last  
System.out.print("Please Enter Author's First Name: ");  
String First **=** sc.nextLine();

//Value gets stored in First  
**for**(Author A **:** addA.getAuthor()) {  
//This takes in the author's full name, disregarding cap sensitivity, and prints out all the books and book info that are under that author  
**if**(A.getLast().equalsIgnoreCase(Last) **&&** A.getFirst().equalsIgnoreCase(First)) {  
System.out.println(A);  
check **=** 1;  
}//end if  
}//end for   
//since the check was not changed that means that the book wasn't in the system.  
//the book will now add the Author and Book information to the system   
**if**(check **==** 0){  
System.out.println("Sorry but that Author/Book isn't in our library. Lets order it now");  
//Retrives the Last name of the author and assigns it to variable Last  
System.out.print("What is the Author's Last Name: ");  
Last **=** sc.nextLine();  
//Retrives the First name of the author and assigns it to variable First  
System.out.print("What is the Author's First Name: ");  
First **=** sc.nextLine();  
//Creates a temp holder for the Variables Last,First used in the last step to be input later with the appropriate book information  
Author temp **=** **new** Author(Last,First);  
//Adding the author to the array from the temp using .addAuthor  
**if**(addA.addAuthor(temp)) {  
**do** {  
System.out.print("Please Enter the Book Title you wish to Order: ");  
String Title **=** sc.nextLine();  
//Retrieves the Title from the User's Input and stores it in Title  
System.out.print("Please Enter the Book's ISBN: ");  
String ISBN **=** sc.nextLine();  
//Retrives the ISBN from the User's Input and stores it in ISBN  
System.out.print("What is it's Genre: ");  
String Genre **=** sc.nextLine();  
//Retrives the Genre from the User's Input and stores it in Genre  
System.out.print("What is the Price of the Book: ");  
**double** Price **=** sc.nextDouble();  
//Retrives the Price from the User's Input and stores it in Price   
sc.nextLine();  
//Adds the Book information to the corresponding Author  
**if**(**!**temp.addBook(**new** Book(Title,ISBN,Genre,Price))){  
**break**;  
}//end if  
//prompts user if they would like to add more books/authors by entering either yes or no.  
System.out.print("Would You Like to Add more Books? Please enter yes or no: ");  
selection **=** sc.nextLine();  
}//end do  
//this allows the user to enter it without having to worry about spelling it in Uppercase or Lowercase  
**while**(selection.equalsIgnoreCase("yes"));

}//end if  
//if the author has 5 books than this will be prompted to the user, and the information will not be added  
**else**{  
System.out.println("Sorry, The Author Can Only Have 5 Books in the Storage");  
}//end else  
}//end if  
}//End else if  
//if no is selected or if the user chose to print, the user will be promted with the selection of what to do next.  
System.out.print("Press 1 to Print all, Press 2 to Search for an Author, Press 3 to Exit the program:\n");

option **=** sc.nextInt();

//allows for the user to move on to the next selection  
}//End While  
}//End Main Method  
}//End Class

**)4.2 Program and Module Description**

Main

The main function creates the programmed books, authors, and carries out the function of adding, searching, and showing all the books. It also gives you the option to exit, at the end of the action.

**5. Solution Testing**

Test the program with following data domain:

The Authors stored in the file are Joanne Rowling, Jerome Salinger, Susan Hinton, Lois Lowry.

Test the program with following data:  Input 2 then:

Please Enter Author's Last Name: **rowling**  
Please Enter Author's First Name: **joanne**  
Rowling, Joanne  
Book: [Book Title: Harry Potter and the Sorcerer's Stone  
ISBN: 465498465  
Genre: Fantasy Fiction  
Price: 8.99  
etc…

**6. Testing: Output**

//Description: This program will create a library of

//books and authors for the user to see/choose from

//initially. Upon Running it will prompt the user to

//perform an action (be it a search or print). Add

//the author and book info if not in the system and //author caps aren’t reached.

Enter '1': to Show all authors   
 Enter '2': to Search for an Author  
 Enter '3': to Exit

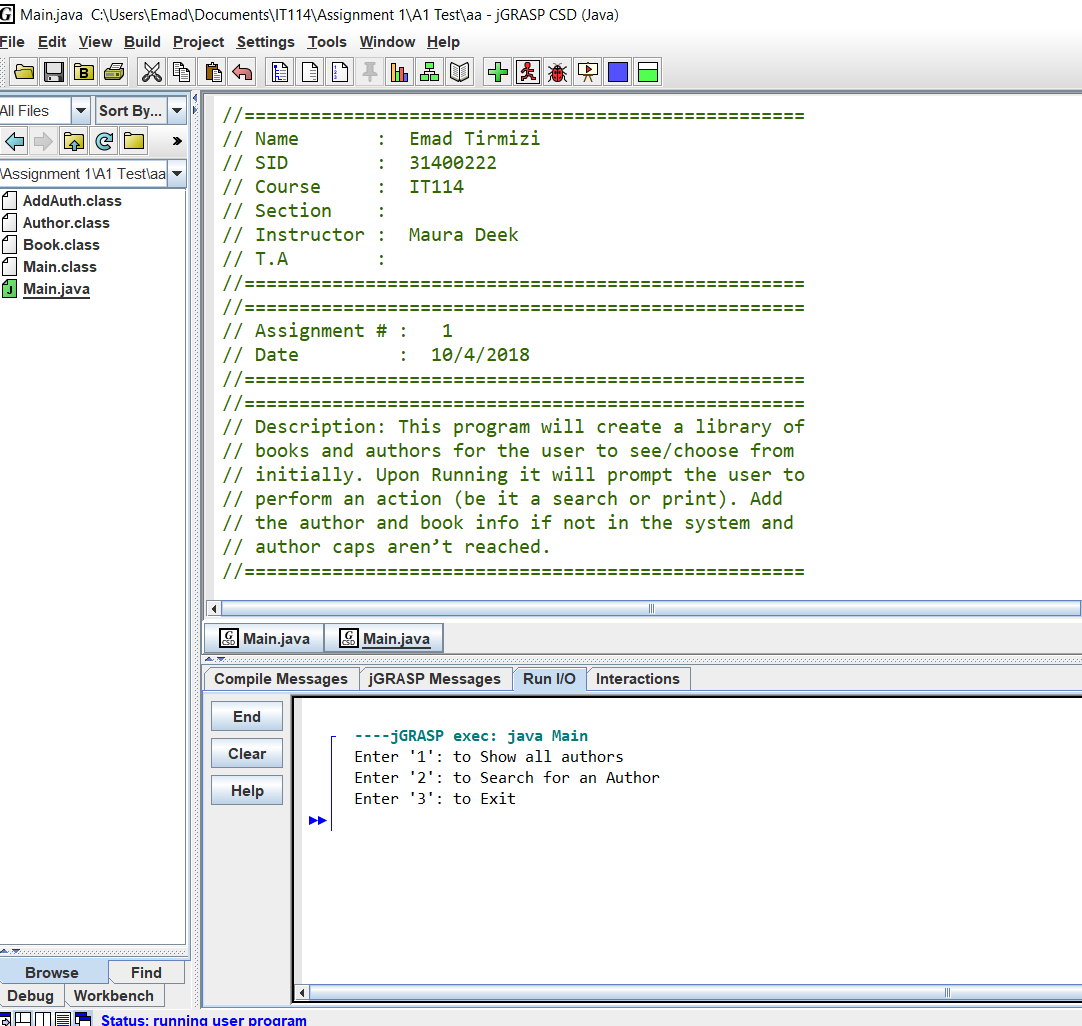
1  
Salinger, Jerome  
Book: [Book Title: Catcher in the Rye  
ISBN: 11511151  
Genre: Literary Realism  
Price: 4.93  
  
]  
Hinton, Susan  
Book: [Book Title: The Outsiders  
ISBN: 1212154  
Genre: Young Adult Fiction  
Price: 14.0  
etc…

Press 1 to Print all, Press 2 to Search for an Author, Press 3 to Exit the program:  
2  
Please Enter Author's Last Name: rowling  
Please Enter Author's First Name: joanne  
Rowling, Joanne  
Book: [Book Title: Harry Potter and the Sorcerer's Stone  
ISBN: 465498465  
Genre: Fantasy Fiction  
Price: 8.99  
  
, Book Title: Harry Potter and the Chamber of Secrets  
ISBN: 4194119  
Genre: Young Adult Fiction  
Price: 8.99  
etc…

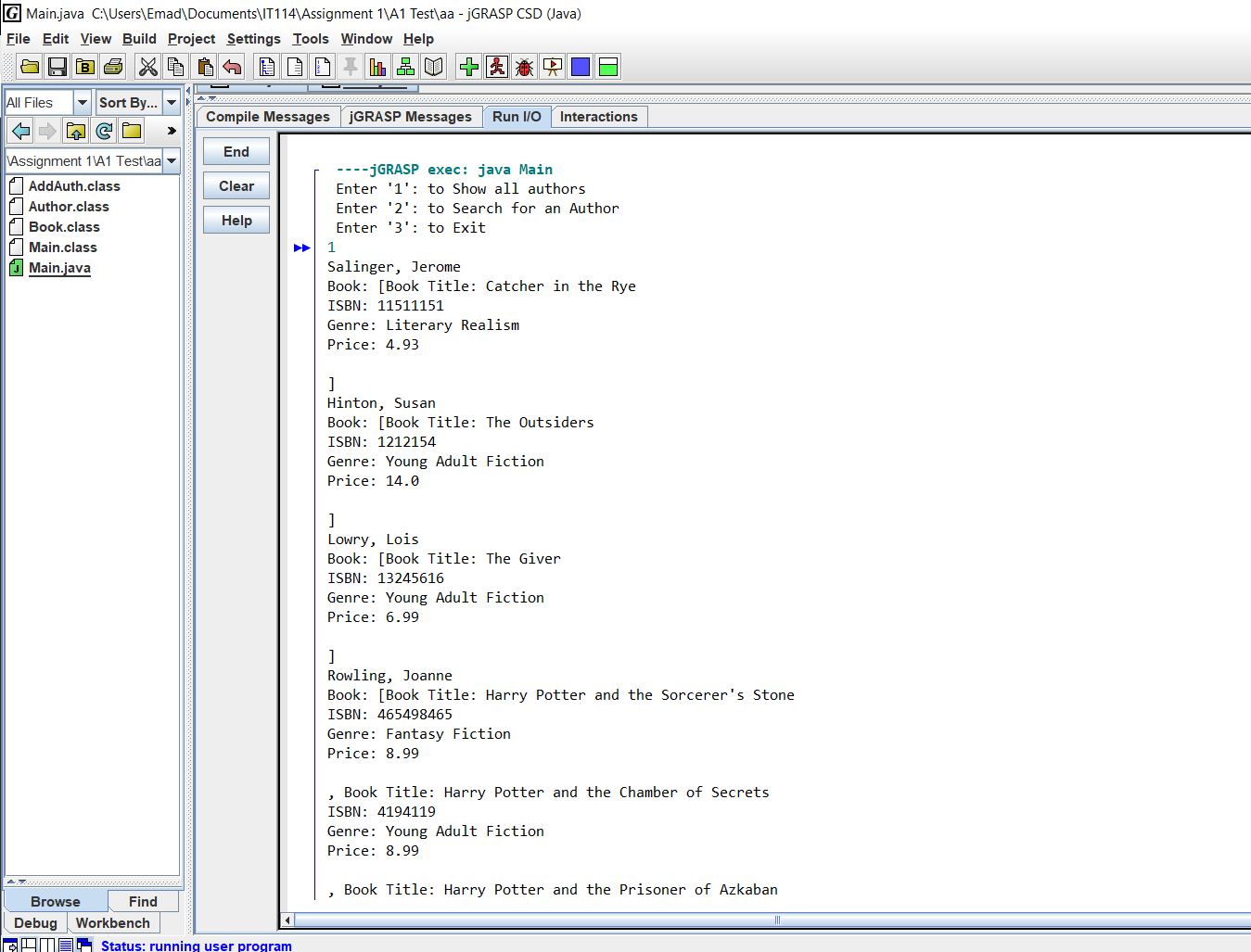
Press 1 to Print all, Press 2 to Search for an Author, Press 3 to Exit the program:  
2  
Please Enter Author's Last Name: me  
Please Enter Author's First Name: my  
Sorry but that Author/Book isn't in our library. Lets order it now  
What is the Author's Last Name: mo  
What is the Author's First Name: me  
Please Enter the Book Title you wish to Order: maw  
Please Enter the Book's ISBN: 12345  
What is it's Genre: Educational  
What is the Price of the Book: 1000.00  
Would You Like to Add more Books? Please enter yes or no: yes  
Please Enter the Book Title you wish to Order: works  
Please Enter the Book's ISBN: 12355  
What is it's Genre: Educational  
What is the Price of the Book: 2.00  
Would You Like to Add more Books? Please enter yes or no: no  
Press 1 to Print all, Press 2 to Search for an Author, Press 3 to Exit the program:  
1  
Salinger, Jerome  
Book: [Book Title: Catcher in the Rye  
ISBN: 11511151  
Genre: Literary Realism  
Price: 4.93  
  
]  
Hinton, Susan  
Book: [Book Title: The Outsiders  
ISBN: 1212154  
Genre: Young Adult Fiction  
Price: 14.0  
  
]  
Lowry, Lois  
Book: [Book Title: The Giver  
ISBN: 13245616  
Genre: Young Adult Fiction  
Price: 6.99  
  
]  
Rowling, Joanne  
Book: [Book Title: Harry Potter and the Sorcerer's Stone  
ISBN: 465498465  
Genre: Fantasy Fiction  
Price: 8.99  
  
, Book Title: Harry Potter and the Chamber of Secrets  
ISBN: 4194119  
Genre: Young Adult Fiction  
Price: 8.99  
  
, Book Title: Harry Potter and the Prisoner of Azkaban  
ISBN: 559121  
Genre: Young Adult Fiction  
Price: 8.99  
  
, Book Title: Harry Potter and the Goblet of Fire  
ISBN: 131881  
Genre: Young Adult Fiction  
Price: 8.99  
  
, Book Title: Harry Potter and the Order of the Phoenix  
ISBN: 33652  
Genre: Young Adult Fiction  
Price: 8.99  
  
]  
mo, me  
Book: [Book Title: maw  
ISBN: 12345  
Genre: Educational  
Price: 1000.0  
  
, Book Title: works  
ISBN: 12355  
Genre: Educational  
Price: 2.0  
  
]  
Press 1 to Print all, Press 2 to Search for an Author, Press 3 to Exit the program:  
3  
  
 **----jGRASP: operation complete.**

**(This is just a sample what you need to submit are actual screenshots of I/O or files)**

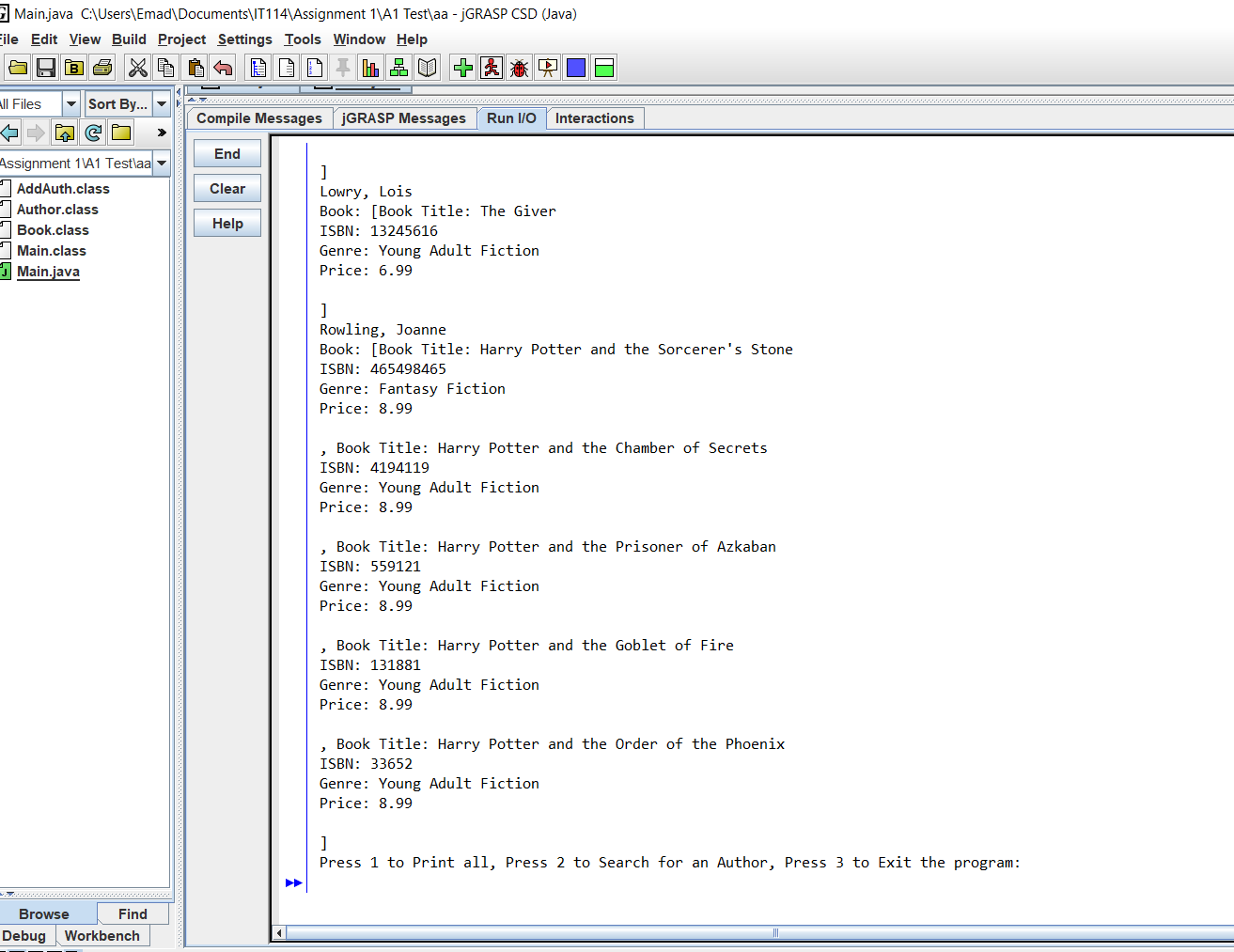
Selection:



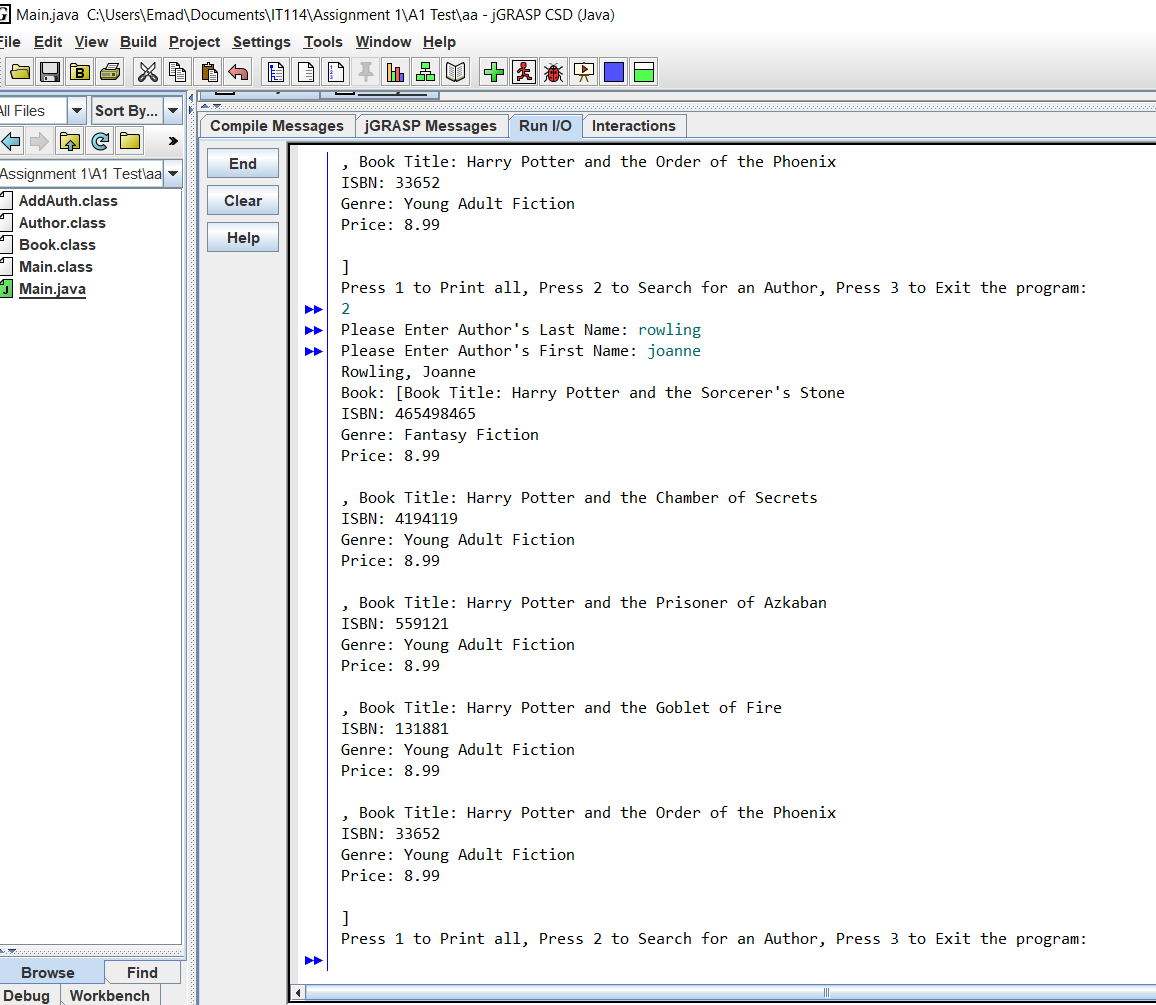
Show All:



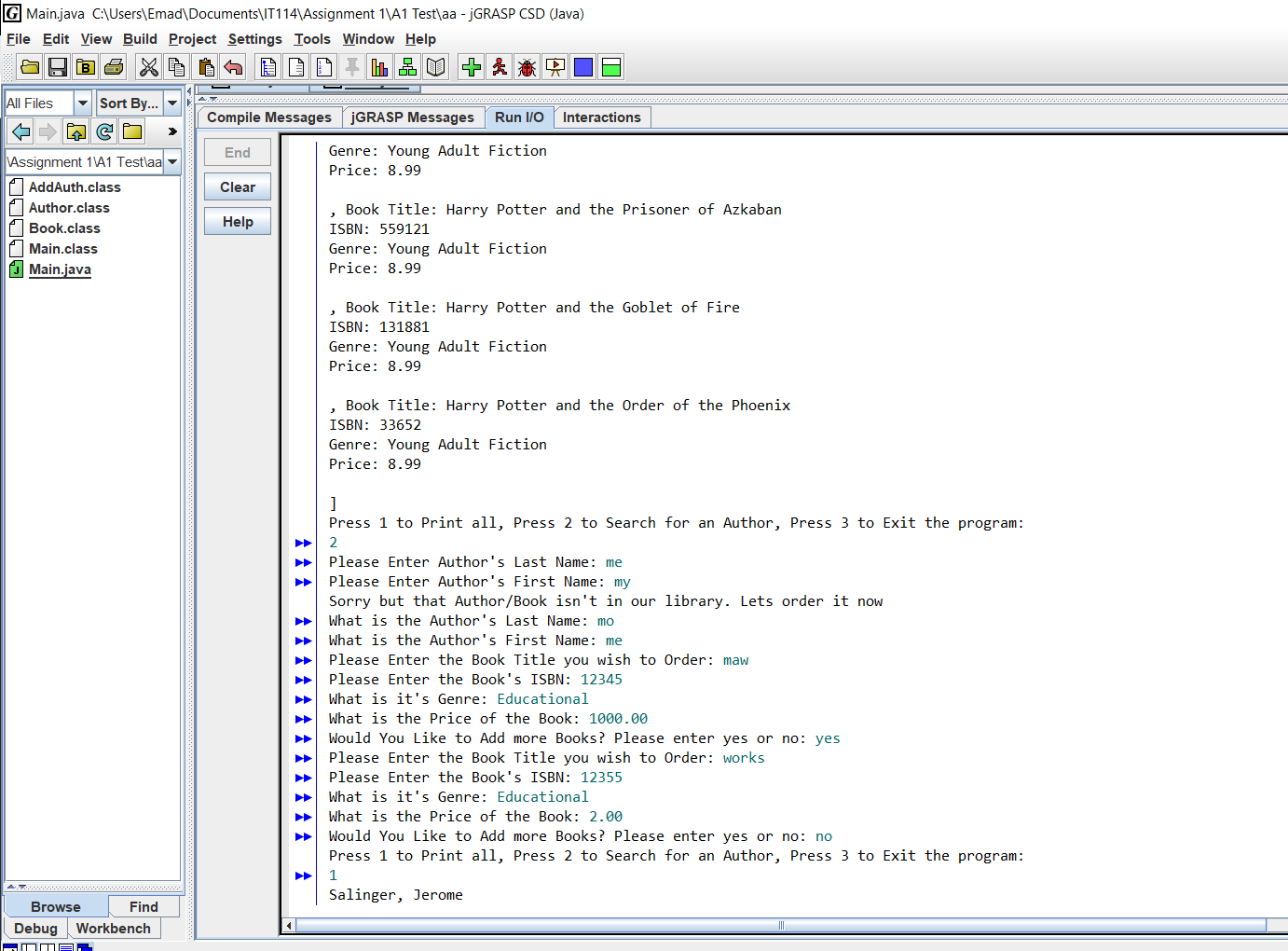
Show all Cont:



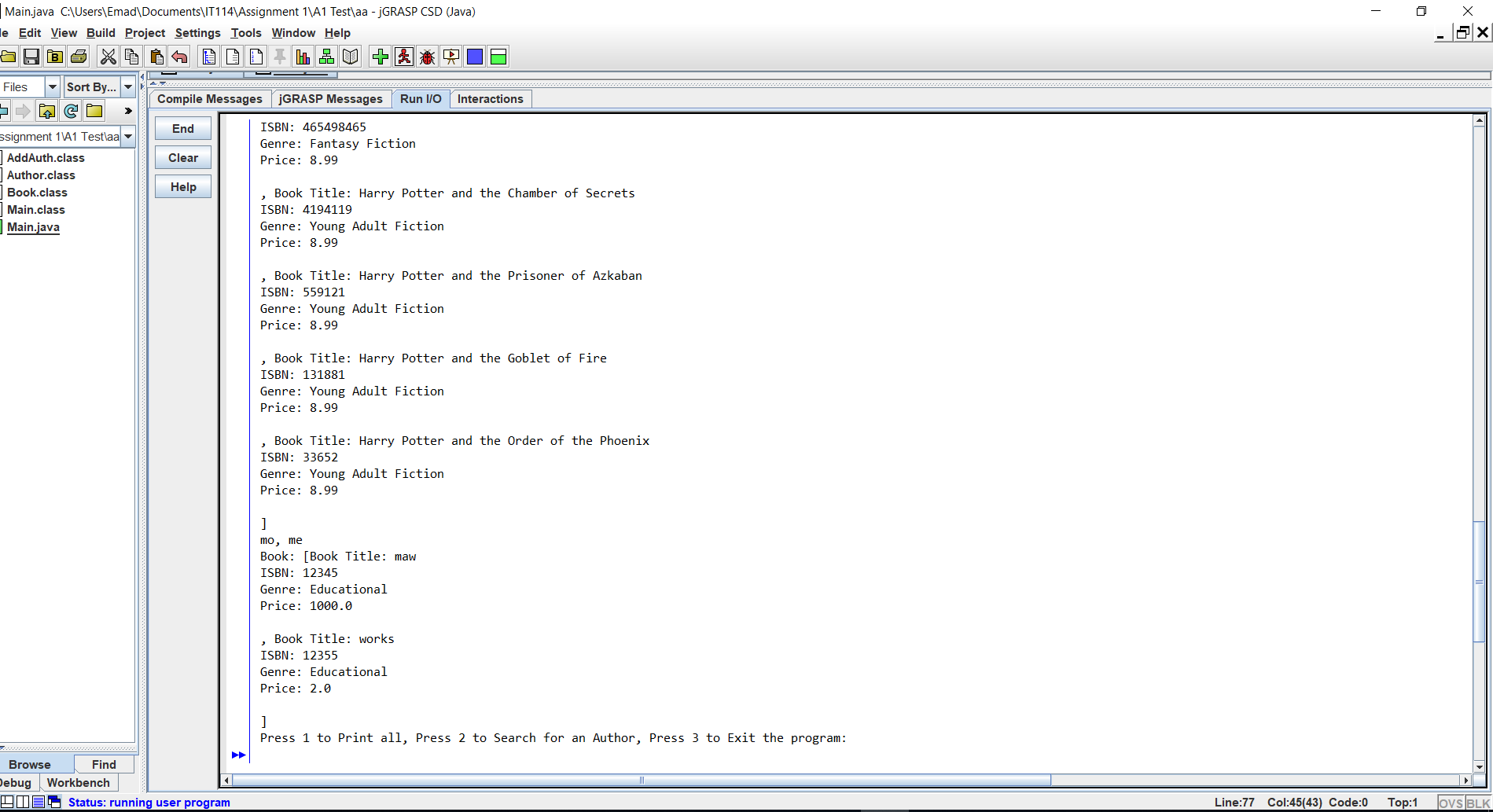
Searching for Specific Author:



Author not Found/Adding Author and Book Info:



Added Items show in Search:



Exit Function:

