

SCHOOL OF COMPUTING
DIPLOMA IN INFORMATION TECHNOLOGY
ST0509 JAVA PROGRAMMING
2025/2026 SEMESTER 1
PRACTICAL ASSIGNMENT 1

Objective of Assignment

To allow students to practice what they have learnt in the module by developing a Java program to simulate a student management system.

Instructions and Guidelines:

1. The assignment should be done in a group of 2 and will account for **30%** of your final grade.
2. The assignment should be submitted by **Monday, 30 Jun, 2024 8:00 am**.
3. The development platform will be in Java using NetBeans IDE.
4. The interview will be conducted during the practical lessons on week 11. You are expected to explain the program logic and modify the program during the interview. If you are absent from the interview, **you will be awarded zero mark for the assignment.**
5. **No marks will be awarded**, if the work is copied or you have allowed others to copy your work.
6. **50%** of the marks will be deducted for assignments that are received within **ONE (1) calendar** day after the submission deadline. No marks will be given thereafter.
7. Prepare the Declaration of Academic Integrity (SOC) form. Your Assignment will NOT be marked if the form is not submitted. The form is available in BrightSpace under Assignments >Forms > Declaration of Academic Integrity (SOC) form.

Warning: Plagiarism means passing off as one's own the ideas, works, writings, etc., which belong to another person. In accordance with this definition, you are committing plagiarism if you copy the work of another person and turning it in as your own, even if you would have the permission of that person. Plagiarism is a serious offence and disciplinary action will be taken against you. If you are guilty of plagiarism, you may fail all modules in the semester, or even be liable for expulsion.

Overview of the system

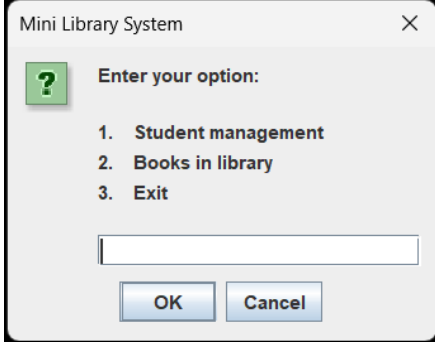
You are tasked to develop a Java program to simulate a mini Library Management System.

There will be 2 parts, a student management system and book management system. Each member will do one part of the system and will be graded for their work on each part of the system.

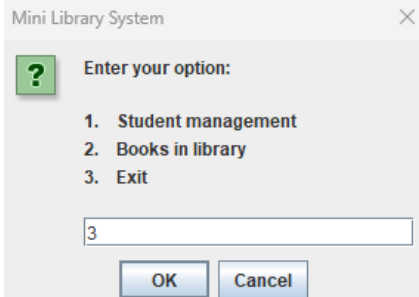
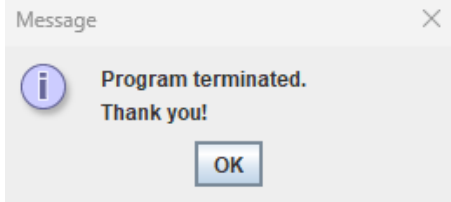
Note that proper validation should be done to check users enter valid values relating to numeric values entry.

Member A requirements:

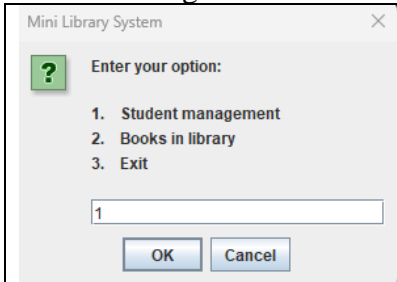
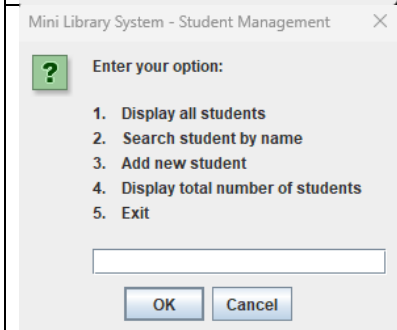
Main Menu Interface Page for Mini Library System.

	<p>Option 1 – Student management Student A will complete student management.</p> <p>Option 2 – Books in library Student will complete books in library.</p> <p>Option 3 – Exit Java Program Terminate</p>
--	---

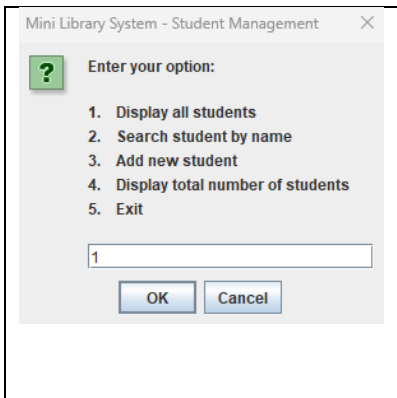
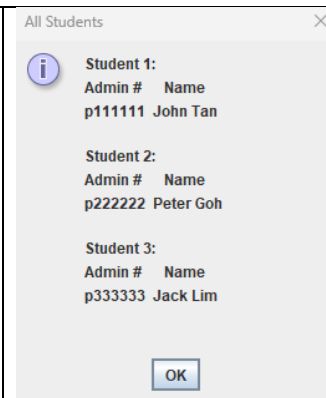
Option 3 – Exit

	 <p>Java Programm terminated.</p>
---	---

Student Management

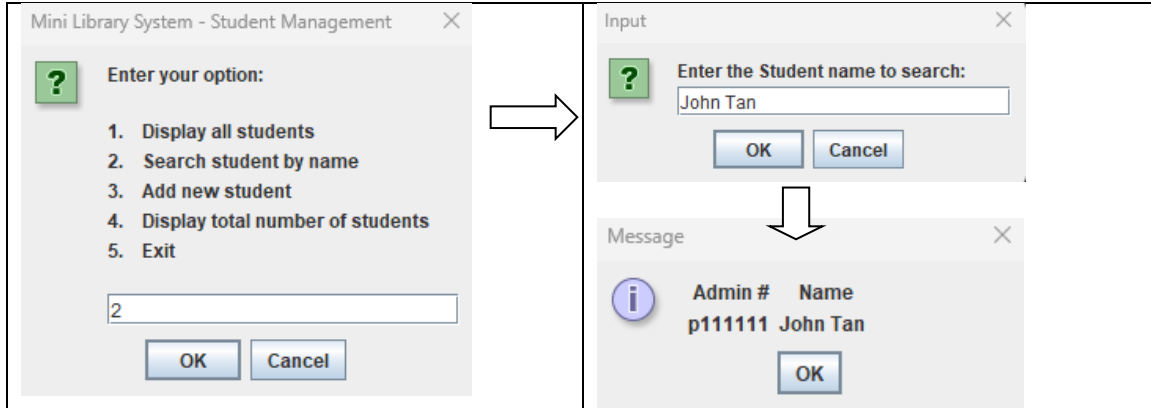
 <p>Mini Library System</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Student management 2. Books in library 3. Exit <input type="text" value="1"/> <p>OK Cancel</p>	Option 1 – Student management
 <p>Mini Library System - Student Management</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Display all students 2. Search student by name 3. Add new student 4. Display total number of students 5. Exit <input type="text"/> <p>OK Cancel</p>	Student Management Menu

Option 1: Display all students

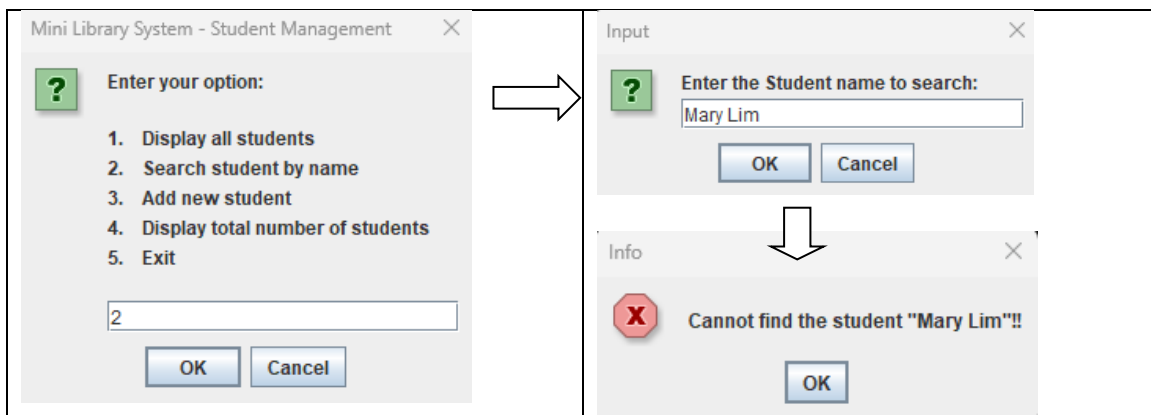
 <p>Mini Library System - Student Management</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Display all students 2. Search student by name 3. Add new student 4. Display total number of students 5. Exit <input type="text" value="1"/> <p>OK Cancel</p>	 <p>All Students</p> <p>Student 1: Admin # Name p111111 John Tan</p> <p>Student 2: Admin # Name p222222 Peter Goh</p> <p>Student 3: Admin # Name p333333 Jack Lim</p> <p>OK</p>
---	--

If the user chooses option 1, the program displays all students' admin number and name.

Option 2: Search student by name

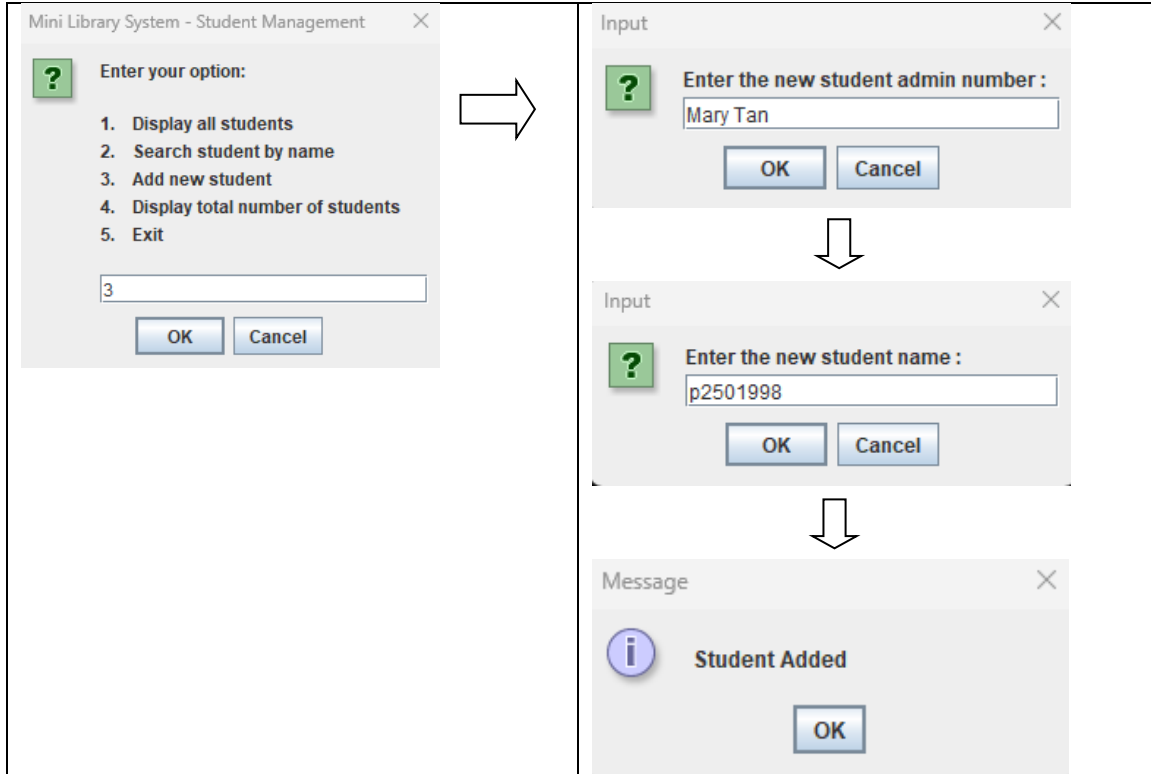


If the user chooses option 2 to search for a student, the program prompts the user to enter the name of student to search and displays the result accordingly. It should display the admin number and name. For this program, we will assume name is unique within the application.



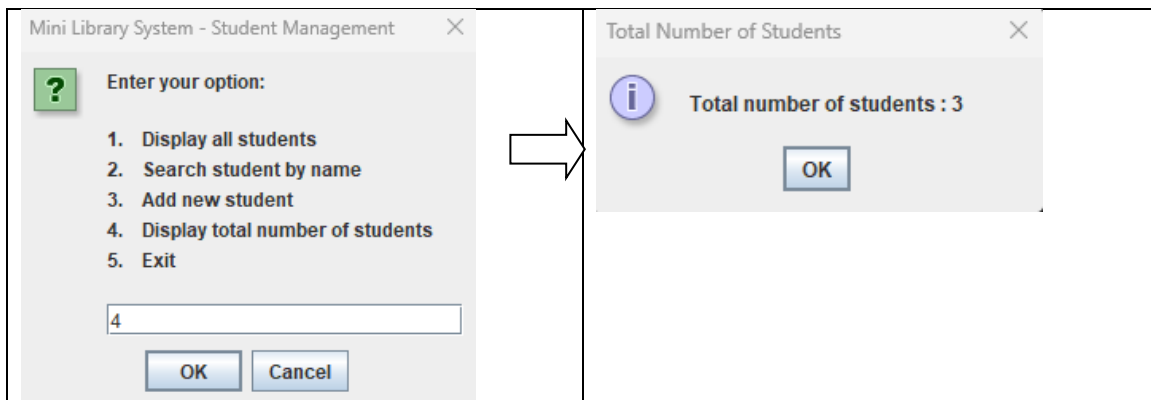
If student name is not found, the program will show cannot find the student “<student name>”.

Option 3: Add new student



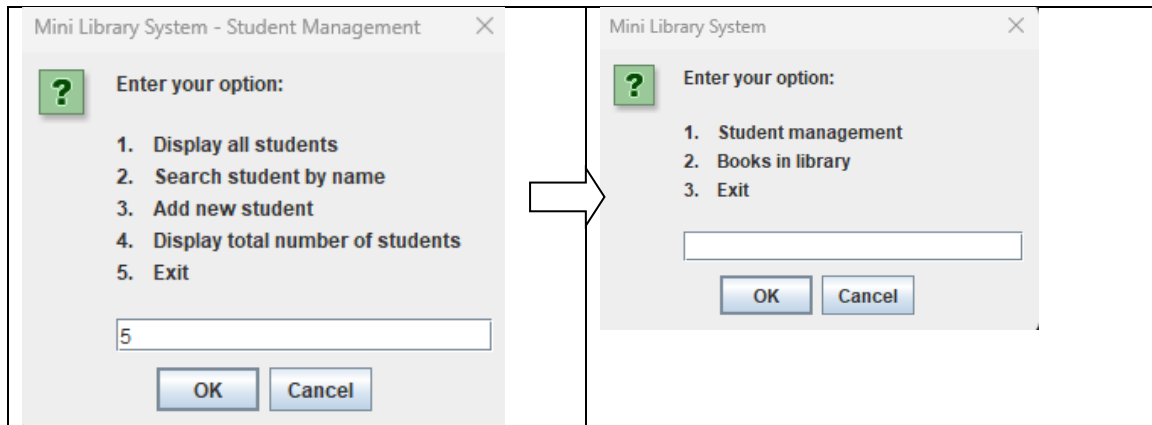
If the user chooses option 3 to add new student, the program prompts the user to enter the new student name and admin number. Prompt user that the student is successfully added.

Option 4: Display total number of students



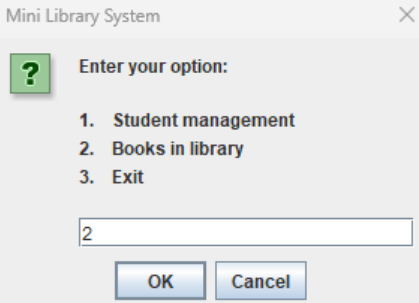
If the user chooses option 4 to display total number of students, the program will compute the total number of students and display in the message dialog box.

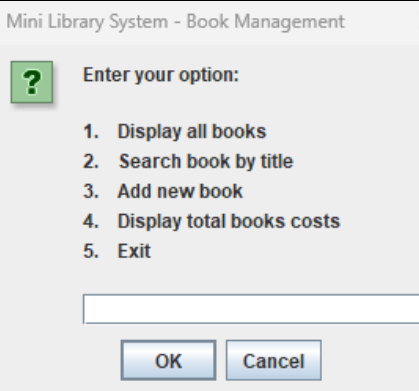
Option 5: Exit



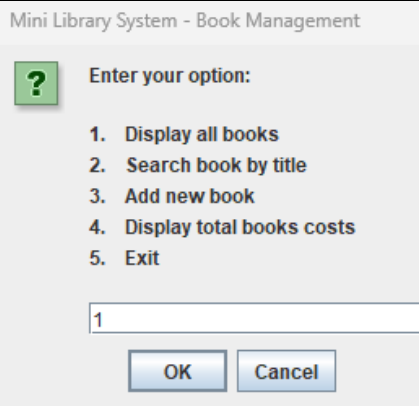
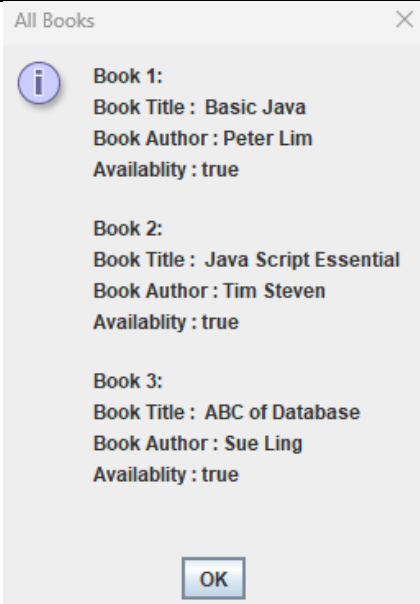
If the user chooses option 5 Exit, the program return back to the main menu of the Mini Library System.

Member B Requirements:**Book in library**

 <p>Mini Library System</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Student management 2. Books in library 3. Exit <p>2</p> <p>OK Cancel</p>	<p>Option 2 - Books in library</p>
--	------------------------------------

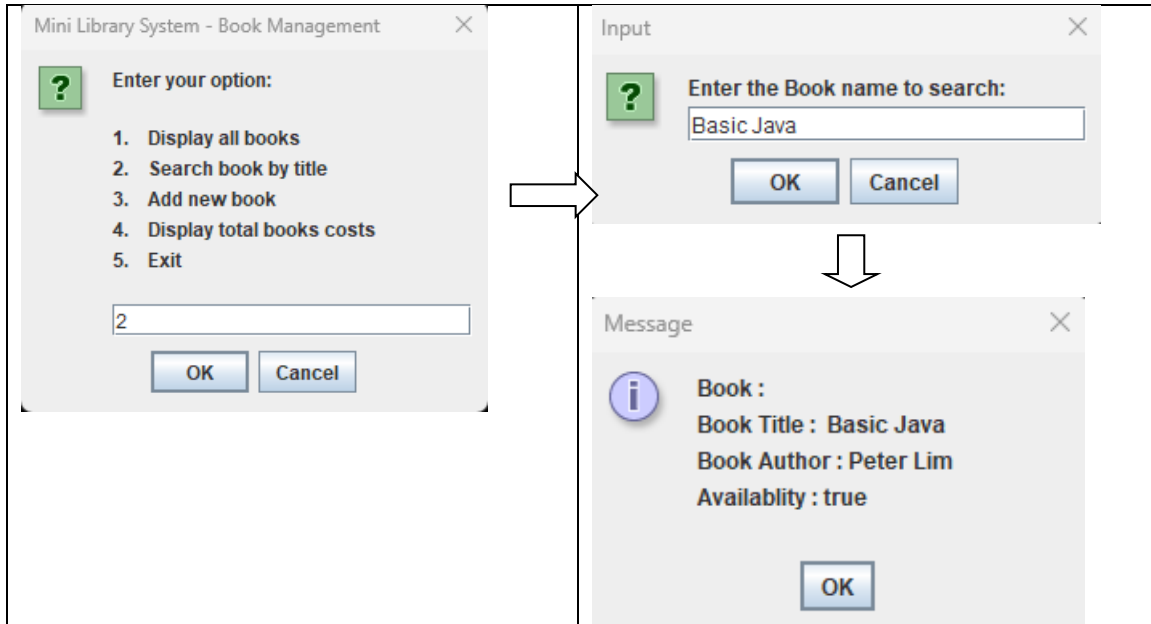
 <p>Mini Library System - Book Management</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Display all books 2. Search book by title 3. Add new book 4. Display total books costs 5. Exit <p></p> <p>OK Cancel</p>	<p>Book Management Menu</p>
--	-----------------------------

Option 1: Display all books

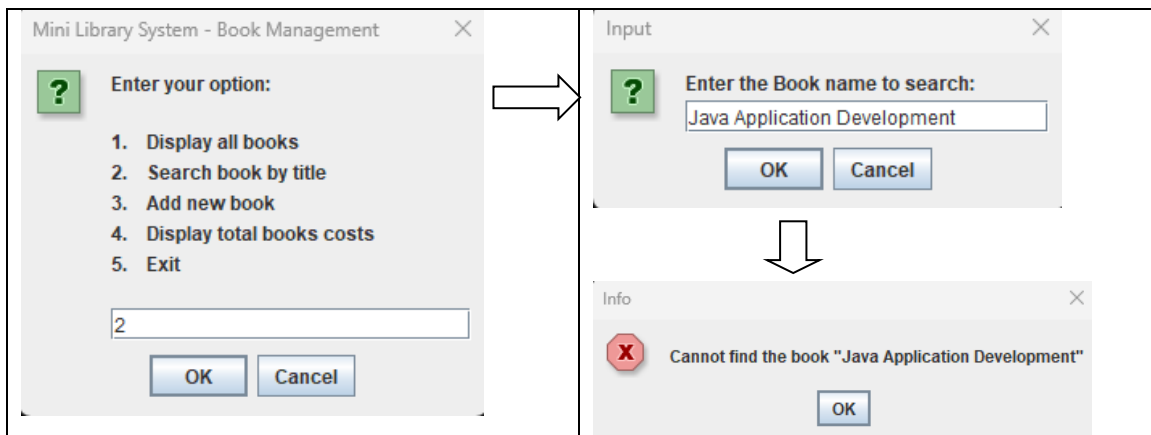
 <p>Mini Library System - Book Management</p> <p>Enter your option:</p> <ol style="list-style-type: none"> 1. Display all books 2. Search book by title 3. Add new book 4. Display total books costs 5. Exit <p>1</p> <p>OK Cancel</p>	 <p>All Books</p> <p>Book 1: Book Title : Basic Java Book Author : Peter Lim Availability : true</p> <p>Book 2: Book Title : Java Script Essential Book Author : Tim Steven Availability : true</p> <p>Book 3: Book Title : ABC of Database Book Author : Sue Ling Availability : true</p> <p>OK</p>
--	--

If the user chooses option 1, the program displays all books' title, author and availability.

Option 2: Search book by title

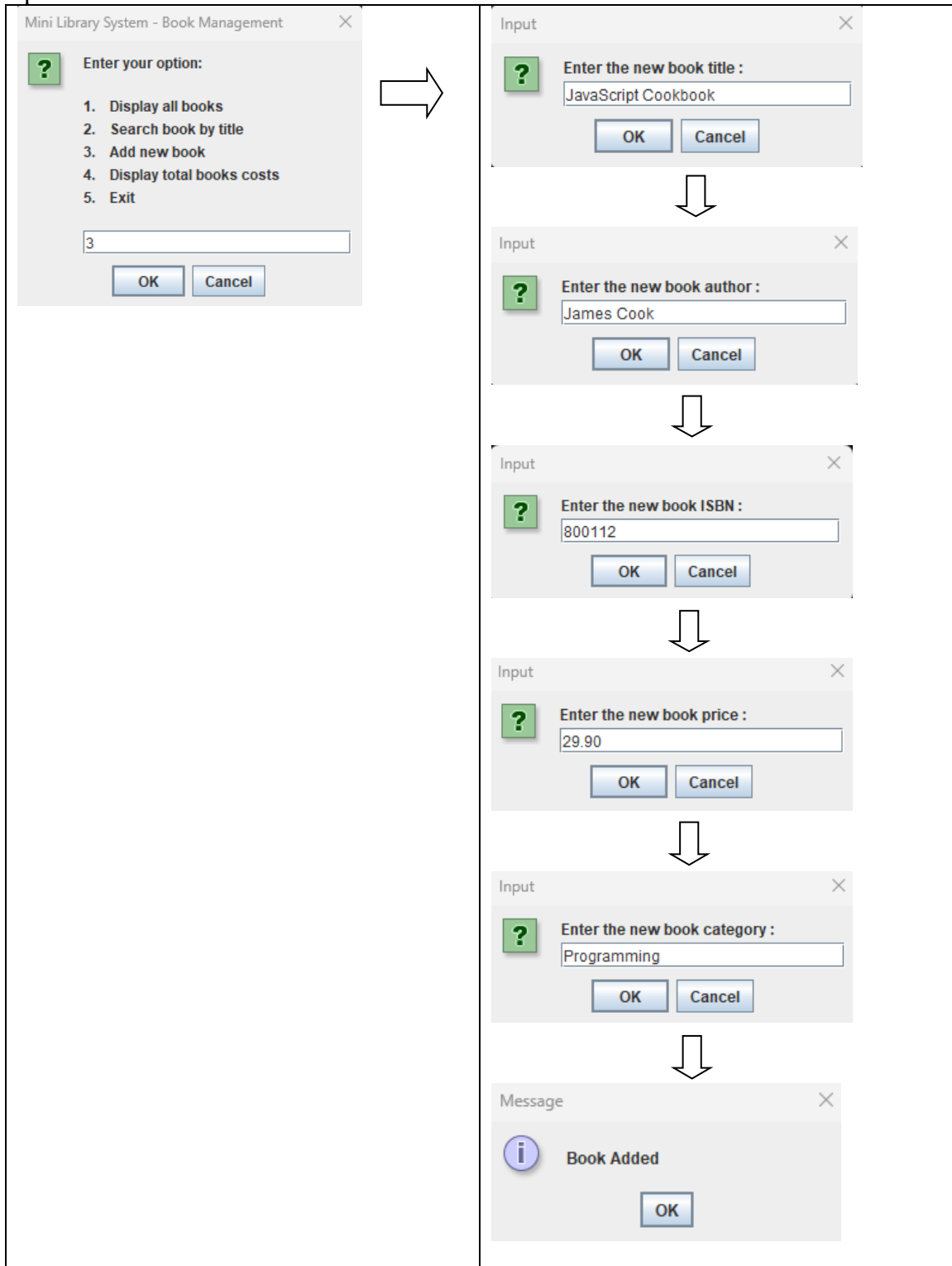


If the user chooses option 2 to search book by title, the program prompts the user to enter the book title to search and displays the result accordingly. It should display the book title, author and availability. For this program, we will assume book title is unique within the application.



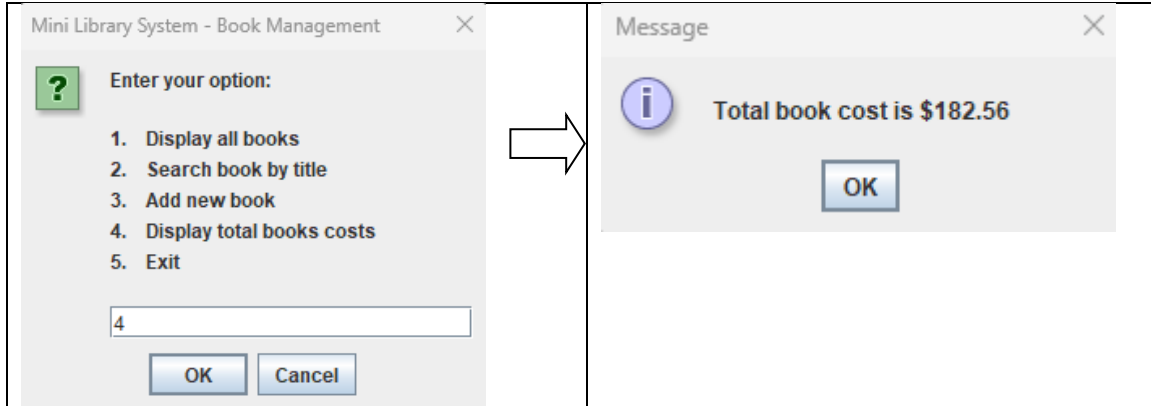
If book title is not found, the program will show cannot find the book "<book title>".

Option 3: Add new book



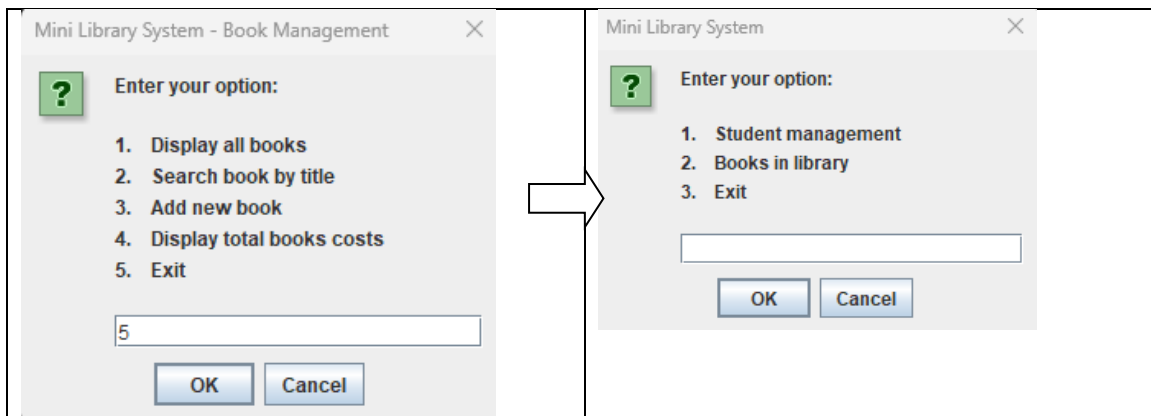
If the user chooses option 3 to add new book, the program prompts the user to enter the new book title, book author, book ISBN, book price and book category. Prompt user that the book is successfully added.

Option 4: Display total book costs



If the user chooses option 4 to display total books costs, the program will compute the total price of books and display in the message dialog box.

Option 5: Exit



If the user chooses option 5 Exit, the program return back to the main menu of the Mini Library System.

Basic Requirements

Write the Java programs using appropriate methods, objects and arrays to perform the tasks stated in **Overview of the system** section.

1. Write a Java class ***Book*** (or similar class) to represent book object, you should have instance variables to store the book title, author, ISBN, price, category and available for loan.
Add a constructor, appropriate **get** or **set** methods, and any other methods to the class if necessary.
2. Write a Java class ***Student*** (or similar class) to represent a student object, you should have instance variables to store the admin number and name and the **Books** taken by a student. Kindly take note that a student will borrow more than one books.
Add a constructor, appropriate **get** or **set** methods, or any other methods to the class if necessary, example an instance method to count the total number of student.
3. Write a Java class ***StudentManagement*** (or similar class) to manage the students. The class should have methods that will be invoked by the ***main*** method such as to create students, display students, search student, compute total number of student,, etc. **In this class, please create at least 8 Student objects and store in an array.**
4. Write a Java class ***BookManagement*** (or similar class) to manage the books. The class should have methods that will be invoked by the ***main*** method such as to create new book, display books, search book by title, compute total costs of books,, etc. **In this class, please create at least 5 Book objects and store in an array.**
5. Write a Java class ***StudenLibrary*** (or similar class) that contains the ***main()*** method for the entire application. The ***main()*** method is where your program will start to run. It displays the menu and calls the methods in ***StudentManagement*** class and ***BooksManagement*** class accordingly.
6. Implement the advanced features such as, but not limited to the followings:
 - Display students' information in a tabulate format.
 - Sound effects.
 - Any other features that enhance the system.

Note: Do NOT implement a full Graphic User Interface (GUI) program for CA1 as this will be the requirement for CA2. For the purpose of CA1, you should be using the dialogs available from the ***JOptionPane*** class.

7. Zip the complete NetBeans project of your CA1 and submit to BrightSpace. **Provide your Class, Admission Number and Name on each of the Java classes.**

Please keep in mind that advanced features are just bonus features. The main bulk of marks are allocated to the completion of a workable program that meets the minimum requirements. You should try to fulfill the minimum requirements before you attempt to include any advance features.

Assessment Guidelines

The assignment will be assessed based on the following criteria:

- Ability to demonstrate the minimum requirements of the system
- Program design such as:
 - Correct and efficient usage of classes and programming constructs
 - Appropriate method decomposition
 - Appropriate validations
 - Code efficiency
- Program readability:
 - Meaningful identifiers
 - Meaningful comments and indentation in source code
- Innovation and creativity, and/or any advanced features
- Independent work and the understanding of the concepts & methodology
- Question & Answer during interview

Suggested data:

Students

Attribute
Name
Admin Number
Books[]

Books

Attribute
Title
Author
ISBN
Price
Category
Available

-- End --