

# WRAV102/MSEV102: Practical 9

*Complete these tasks, and show completion during your allocated session*

## Objectives

- Classes and Objects
- Text files
- List class
- Arraylist
- Sorting using bubble sort, selection sort and insertion sort algorithms
- Using the sorted state of the list to determine the appropriate searching algorithm – linear search or binary search.

## How to save your program for later use?

1. Use the **File/Save All** option. This saves it in the location you identified in the beginning.

## How do I submit my work?

Show your completed tasks to an assistant on duty before the end of the session. The marking rubric that will be used is on page 3 of this document.

## Important note:

When working with bigger projects, it is good practice to clean any temporary files from your project folder. To do this in Visual Studio, click on Build, then select Clean Solution. Save your projects on your Submissions folder (S drive folder).

## Task

Imagine you are tasked with creating a simple library catalogue system. You need to create a program that can store information about books, including their ISBN number, title, author, and year of publication. Users should be able to add new books to the catalogue, delete books, search for and then display all information on a specific book (based on the ISBN number), list all the books in the catalogue, and sort the catalogue in ascending order of ISBN number, Author or year of publication.

For this task, a starter project has been created and is available on Moodle. The starter project is compressed to a zipped format. Once you have downloaded the project folder, you need to unzip the folder first.

- Right-click on the file `Library.zip`
- Select Extract All
- Browse to the folder where you would like the Library folder to be (I would suggest putting it into a Prac 9 folder)
- Click on Extract

You should follow one of the following methods to continue with this task:

- Open the application (*by opening the .sln file*), complete the task by checking the requirements below. The text file is already in the correct folder.
- Create a new console application (*select the option to create a new Console App (.NET Framework)*). Create the required classes in your application. Copy the provided code from the starter application to the relevant classes of your newly created application. You will also need to copy the provided text file to the bin\Debug folder of your project folder.

The skeleton project contains:

- An object class named **Book**
- A list class named **LibraryCatalog**
- The regular application class (program.cs)

## Book class

The Book class already contains some aspects, do not alter the existing code, but add any methods required for the functionality of the program.

A book object has the following properties: ISBN number, Title, Author and Publication Year

## LibraryCatalog class

The LibraryCatalog class already contains some aspects, do not alter the existing code, but add methods to do the following:

- Initialise the arraylist from the provided text file (BookData.txt).
- Search for a book object, based on an ISBN number provided – depending on the sorted state of the list, you should determine whether to make use of a linear search or binary search algorithm (both search algorithms should be provided for). You should use binary search if the list is sorted on ISBN number at the time of the request. *For the sake of this task, include a feedback message indicating whether the process is making use of the linear or binary search algorithm.*
- Bubble sort to sort the book list in ascending order of ISBN number.
- Selection sort to sort the book list in ascending order of Author.
- Insertion sort to sort the book list in ascending order of Publication year.
- Add a new book to the end of the list.

For the duration of this task, keep track of sorted state of the list as follows:

- List not sorted: sortedState = 0
- List sorted in ascending order of ISBN number: sortedState = 1
- List sorted in ascending order of Author: sortedState = 2
- List sorted in ascending order of Publication year: sortedState = 3

Make use of private methods to provide the implementation of the sorting and searching algorithms, with public methods that will call to these private methods, and set the sorted state as required – the public methods is what you should call from the application class.

## Application class

The main method allows the user to interact with the library catalogue. The application class already contains some aspects, do not alter the existing code, but add methods as needed to provide the required functionality:

Functionality required:

- Allow the user to add a new book to the list. Prompt the user for the all the required values. Display a confirmation message when the book was added to the list.
- Allow the user to delete a book from the list. Display appropriate messages.
  - Prompts the user for an ISBN number and display all the data for the specific book. If the book is not found, display an appropriate message.
  - List all books.
  - Sort the list in ascending order of ISBN number, then display the list. (*You should use the bubble sort algorithm for this task.*)
  - Sort the list in ascending order of Author, then display the list. (*You should use the selection sort algorithm for this task.*)
  - Sort the list in ascending order of Publication year, then display the list. (*You should use the insertion sort algorithm for this task.*)

As a last function of your application, you should write the data from the arraylist to a comma delimited file (one line of data for each book). **For the sake of this exercise, write your data to a different file.**

The user should be in control of invoking the sort functionality. You should not call any sorting methods unless that is what the user requested.

### Important Note:

Since you are making use of a list class there should be NO arraylist operations in your Application class; while **all** user input belongs in the application class.

**Prac marking rubric:**

Your prac will be assessed by the assistants on duty (in the actual practical session), based on the following rubric:

	Mark	Option
<b>Book.cs / 2</b> All methods required for the functionality of the program.	0	Does not compile or incomplete.
	2	All of the required methods present and correctly implemented
<b>LibraryCatalog.cs / 5</b>		
Make use of an arraylist to record book objects. All methods required to provide required functionality – but no user input. In addition to methods required for prac 7, the following should be there: As private methods: <ul style="list-style-type: none"> <li>• Bubble sort: sort on ISBN (Asc)</li> <li>• Selection sort: sort on Author (Asc)</li> <li>• Insertion sort: sort on Publication year (Asc)</li> <li>• Binary Search</li> <li>• Linear Search</li> </ul> As public methods: <ul style="list-style-type: none"> <li>• Corresponding methods that can be accessed from the application class (contains method call, and update to sorted state)</li> </ul>	0	Not implemented, or does not compile
	1	Only some of the required methods present
	2	All of the required methods present, but mistakes in implementation
	5	All of the required methods present and correctly implemented
<b>Application class (program.cs) / 5</b>		
Your program should do the following: <ul style="list-style-type: none"> <li>• Declare an instance of class <code>LibraryCatalog</code>.</li> <li>• Populates the list of the text file</li> <li>• Provide the user with a menu of options, from which he can continuously select a task to complete.</li> </ul> Functionality required: <ul style="list-style-type: none"> <li>• Allow the user to add a new book to the list. Prompt the user for the all the required values. Display a confirmation message when the book was added to the list.</li> <li>• Allow the user to delete a book from the list. Display appropriate messages.               <ul style="list-style-type: none"> <li>▪ Prompts the user for an ISBN number and display all the data for the specific book.</li> <li>▪ List all books.</li> <li>▪ Sort the list in ascending order of ISBN number, then display the list. (You should use the bubble sort algorithm for this task.)</li> <li>▪ Sort the list in ascending order of Author, then display the list. (You should use the selection sort algorithm for this task.)</li> <li>▪ Sort the list in ascending order of Publication year, then display the list. (You should use the insertion sort algorithm for this task.)</li> </ul> </li> </ul> As a last function of your application, you should write the data from the arraylist to a comma delimited file (one line of data for each book). For the sake of this exercise, write your data to a different file.	0	Not implemented, or does not compile
	1	Only some of the required methods present
	2	All of the required methods present, but mistakes in implementation
	5	All of the required methods and functionality present and correctly implemented

In addition to the student assistants on duty checking your code and run of the project, they can ask you any number of questions to determine whether you have a good understanding of the code you have submitted. **If you are unable to answer the questions of the assistant, you will not get full marks for the submitted prac tasks, even though it might be running perfectly.**