WRAV102/MSEV102: Practical 8

Complete these tasks, and show completion during your allocated session

Objectives

- Classes and Objects
- Text files
- List class
- Arraylist
- Sorting using bubble sort and selection sort algorithms

Process to complete your tasks

1. Open "Microsoft Visual Studio".
2. Follow the instruction in the slides to **create** a **new console application**. Choose **Windows Console App (.NET Framework)**.
3. Pay attention to where you save your project, so you can easily access it again for submission.

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

The task for prac 8 continuous on the scenario from prac 7 (a program that allows a doctor's staff to work with patient data). When working on your solution, remember to make use of methods as much as possible.

Make a copy of your P7T1 folder, rename it to P8T1 (you do not need to rename any of the files within the project folder). Open this project in VS, and edit /add to your code as required.

At this point your project should contain:

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

A file (`PatientData.txt`) is available on Moodle. Download the file, save to the correct folder, and use it to initialise the patient data.

Patient class

Your patient object class should not need any updates – except for if it was not correct/incomplete for prac 7.

PatientList class

If your PatientList class were incomplete/incorrect for prac 7, please correct it first (if so, check the original instructions of prac 7).

For the prac 8 task, you need to implement the following sorting methods as private methods:

- Bubble sort to sort the patient list in ascending order of patient number
- Selection sort to sort the patient list in ascending order of surname
- Selection sort to sort the patient list in descending order of balance due
- Bubble sort to sort the patient list in descending order of surname

In addition to these private methods, create 4 corresponding public methods that will only contain method calls to the private methods – the public methods is what you should call from the application class.

Note: For prac 7 you were required to provide a method to search through the list (e.g. `FindPatient`), which should return the position of the *wanted* Patient object in the arraylist. The search should be based on the patient number attribute. You also need to provide a method to return a specific object from the arraylist (e.g. `GetPatient`), this method should accept the position of the object in the list, and return the object at that position, if the position is not valid, return the `null` value. **Even though your list might possibly be sorted using one of the newly implement algorithms, you should NOT (at this stage) make changes to your `FindPatient` method – it should still make use of a linear search process when searching for a particular object.**

In your solution, ensure that you use the following concept:

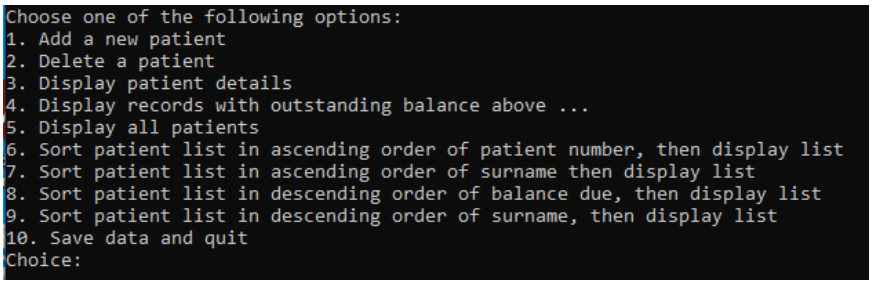
- In the **list class**, the first position in the list is position **0**
- In the **application class**, the first position in the list is position **1**

Application class

Your program should provide the functionality required for prac 7 in addition to the newly adding sorting functionality:

- Declare an instance of class PatientList.
- Populates the list from the text file
- Provide the user with a menu of options, from which he can continuously select a task to complete.

Functionality required:



- Allow the user to add a new Patient to the list. Display a confirmation message when the Patient was added to the list.
- Allow the user to delete a Patient from the list. However, before deleting a Patient from the list, display the current balance of the Patient. Then prompt the user to decide whether to go ahead with the deletion of the Patient. Display appropriate messages.
  - Prompts the user for a patient number and display all the data for the specific record, including the position of the patient in the list (you should make use of a method from the object class to display the patient attributes, and in addition to that, display the position).
  - Prompts the user for a minimum balance due, and displays all the records containing a balance greater than or equal to the entered value.
  - Display all patients.
  - Sort the list in ascending order of patient number, then display the list. (*You should use the bubble sort algorithm for this task.)*
  - Sort the list in ascending order of surname, then display the list. (*You should use the selection sort algorithm for this task.)*
  - Sort the list in descending order of balance due, then display the list. (*You should use the selection sort algorithm for this task.)*
  - Sort the list in descending order of surname, then display the list. (*You should use the bubble 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 patient). **For the sake of this exercise, write your data to a different file.**

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
Patient.cs / 2 Nothing new added to the requirements of prac 7. Private fields for patient number *(make this an integer field)*, surname, initials and current balance owed to the doctor's office. 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
PatientList.cs / 5		
Make use of an arraylist to record patient 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: • `Bubble sort: sort on patient no (Asc)` • `Selection sort: sort on surname (Asc)` • `Selection sort: sort on balance (Desc)` • `Bubble sort: sort on surname (Desc)` As public methods: • 4 Corresponding methods that can be accessed from the application class (only contains method call)	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
Application class (program.cs) **/ 5** Your program should do the following: • Declare an instance of class `PatientList`. • Populates the list of the text file • Provide the user with a menu of options, from which he can continuously select a task to complete. Functionality required: • Allow the user to add a new Patient to the list. Display a confirmation message when the Patient was added to the list. • Allow the user to delete a Patient from the list. However, before deleting a Patient from the list, display the current balance of the Patient. Then prompt the user to decide whether to go ahead with the deletion of the Patient. Display appropriate messages. ▪ Prompts the user for a patient number and display all the data for the specific record, including the position of the patient in the list (you should make use of a method from the object class to display the patient attributes, and in addition to that, display the position). ▪ Prompts the user for a minimum balance due, and displays all the records containing a balance greater than or equal to the entered value. ▪ Display all patients. ▪ Sort the list in ascending order of patient number, then display the list. (*You should use the bubble sort algorithm for this task.)* ▪ Sort the list in ascending order of surname, then display the list. (*You should use the selection sort algorithm for this task.)* ▪ Sort the list in descending order of balance due, then display the list. (*You should use the selection sort algorithm for this task.)* ▪ Sort the list in descending order of surname, then display the list. (*You should use the bubble 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 patient). For the sake of this exercise, write your data to a different file. **In your solution, ensure that you use the following concept:** • In the **list class**, the first position in the list is position **0** • In the **application class**, the first position in the list is position **1**	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.**