|  |  |
| --- | --- |
| **Programming II**  Diploma in IT / CSF / DS  Year 1 (2023/24) Semester 2 | Week 2 |
| **2** hours |
| **Practical 2: Writing C# Programs (Part 2)** | |

**OBJECTIVES**

At the end of this exercise, you should be able to implement C# programs involve

* Generic Collections
* System operations – Date and File I/O

|  |
| --- |
| **IMPORTANT**   * Create a folder, **Week02**, in your hard disk. * For each question, you will be creating a new Console App (.NET Core) project, **Snnnnnnnn\_Question##**, in the **Week02** folder created above *(note:* ***Snnnnnnnn*** *is your Student Number and ## is the question number).* Type the program in the Program class of each project. * At the end of the session, copy the folder **Week02** folder (which contains all your work) to Brightspace. |

Visual Studio has IntelliSense which is extemely useful when writing code. It often gives you help (in the form of a dropdown and greyed out text). To accept the help given, use the <Tab> key twice – once to remove the dropdown and second <Tab> to accept the greyed out text. Try it out when writing the programs!

To learn more about Visual Studio’s IntelliSense, visit: <https://learn.microsoft.com/en-us/visualstudio/ide/using-intellisense?view=vs-2022>

1. My Generic Collections

Write a C# program to:

* 1. Perform the following to an array of size 5:
     1. Prompt the user for 5 integer numbers and add each into the array. Remember to convert the number eneterd by the user to an integer!
     2. Display the contents of the array.
  2. Perform the following to a List:
     1. Copy the contents of the array in (a) to the List.
     2. Prompt the user for more integer numbers and add them to the List. Terminate the loop when the number -1 is entered.
     3. Display the contents of the List.
  3. Perform the following to a Stack:
     1. Copy the contents of the List in (b) to the Stack.
     2. Push the numbers 56 and 72 into the Stack.
     3. Display the contents of the Stack.
     4. Pop the Stack and display the contents of the Stack again.
  4. Perform the following to a Queue:
     1. Copy the contents of the List in (b) to the Queue.
     2. Enqueue the number 23 and 45 into the Queue and display the contents of the Queue.
     3. Dequeue the Queue and display the contents of the Queue again.

1. My Telephone Directory

Create a csv file called PhoneDirectory.csv with comma delimiter that contains your friends’ names and phone numbers.

Write a C# program to:

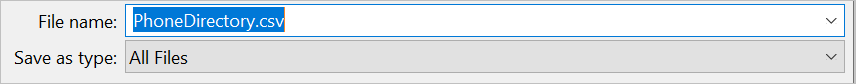
1. Write to the csv file.

Prompts the user for names and phone numbers for an additional list of people and append them to the same csv file. Terminate the loop when the name entered is “Exit”. Display the number of records appended to the file.

Open PhoneDirectory.csv in Notepad to ensure that the new data are added. Determine the difference between “copy always” and “copy if newer”.

Note:

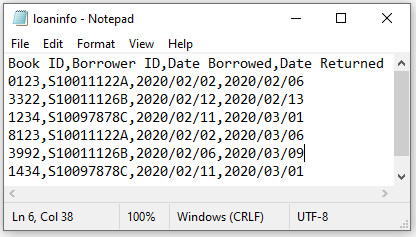
* You may use Notepad to create the file and save it as the given file name (change the “Save as type” to “All Files”).



* Add the data file to your project in Visual Studio.
* Set the property of the file to “copy always” or “copy if newer”. This allows the file to be copied from the project folder into the bin/Debug/net6.0 folder at runtime.
* Refer to Appendix for more details

1. Read from the csv file and store each record into a Dictionary. Display the contents of the Dictionary.
2. Read from the csv file and store each record into a SortedList. Display the contents of the SortedList.
3. Library Loan

The file loaninfo.csv contains the loan information of the library books as follows:

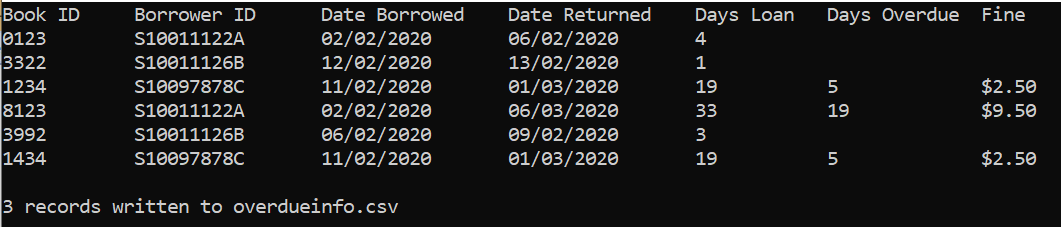


Assume that the maximum loan duration for each book is 14 days. There will be a charge of 50 cents for each day overdue.

Write a C# program to

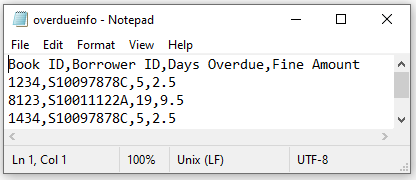
* Read the data from the file and display them together with the number of days the books were loan out. Display also the number of days overdue and fine if the loan duration exceeds 14 days.

Screenshot below shows the sample output:



* Create an output file overdueinfo.csv with book id, borrower id, days overdue and fine amount for books overdue.

Screenshot below shows the sample of overdueinfo.csv:



1. Distance-Based Fares

Distance-Based Fares (DBF) is a fare payment scheme currently used across public buses and MRT/LRT trains in Singapore. Fares are charged based on the total distance travelled (regardless if it is on a bus or train). The distance-based fare calculation is available in the “distance-based-fare.csv” file provided.

Based on the route details of bus service 174 that is available in the “bus\_174.csv” file provided, write a C# program that first displays the route details, then prompts user to enter the boarding bus stop and alighting bus stop. Calculate the distance travelled and the corresponding payable fare.

In addition, given an estimation travel time of 4 minutes per kilometer (km), provide an estimation of the travel time.

The following shows a sample run of the program. The input value is underlined.

|  |
| --- |
| Distance (km) Bus Stop Code Road Bus Stop Description  0 22009 Jurong West Ctrl 3 Boon Lay Int  0.6 21361 Jln Boon Lay Blk 695  0.9 21399 Boon Lay Ave Opp River Valley High Sch  1.3 21421 Boon Lay Ave Blk 176  1.7 21241 Boon Lay Ave Opp Blk 213  1.8 21161 Corporation Rd Opp Jurong JC  3 28169 Jurong Rd Aft Hong Kah Flyover  . . . . .  . . . . .  27.5 06029 Outram Rd Outram Pk Stn  28.1 10011 New Bridge Rd Bef Neil Rd  28.4 10041 Kg Bahru Rd Bef Kampong Bahru Ter  29 10499 Spooner Rd Kampong Bahru Ter  Enter boarding bus stop: 42159  Enter alighting bus stop: 10499  Distance travelled: 18.2km  Fare to pay: $1.65  Estimated duration: 73mins |

Note: Good to split the tasks into 4 methods:

ReadDistanceBasedFare()

ReadBusRoute()

DisplayBusRoute()

ProcessData()

**Appendix**

|  |  |
| --- | --- |
| * Right click on Solutions Explorer > Add > Existing Item |  |
| * Select All files from drop down menu > click on the csv file > Add button. A copy of the csv file will be added to the project folder. |  |
| * Under properties panel, change the drop down to “copy always” or “copy if newer”. |  |
| * During runtime, the csv file will be copied from the project folder into the bin/ Debug/net6.0 folder. | |

**Plagiarism Warning**

**If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical. Disciplinary action may also be taken.**

**Similar action will be taken for student who allows other student(s) to copy his/her work.**