# **CA3 : Due Date: Sunday 23rd April at 21:00**

**The relevant files and a word report must be uploaded to Moodle.**

**Please also use Github as per your lecturer’s instructions.**

## **Problem description**

A **menu driven** program is required to allow the user to explore passenger data from ships who sailed to New York during the 1800s

This data is stored in a CSV file with the following fields ten fields.

|  |  |  |
| --- | --- | --- |
| 1 | LAST NAME |  |
| 2 | FIRST NAME |  |
| 3 | AGE |  |
| 4 | SEX CODE |  |
| 5 | OCCUPATION CODE |  |
| 6 | NATIVE COUNTRY CODE |  |
| 7 | DESTINATION |  |
| 8 | PASSENGER PORT OF EMBARKATION CODE |  |
| 9 | MANIFEST IDENTIFICATION NUMBER | (This is the ship id) |
| 10 | PASSENGER ARRIVAL DATE |  |

The first line of the file will be the headings.

Each subsequent line will contain data about a passenger and the ship they sailed on.

The first seven fields are about an individual passenger.

The last three fields will be the same for each ship.

Moodle contains a file with data for passengers who departed from Donegal to New York at the time of the famine in the mid-1800s. You can use this file to test your program.

## **Detailed Requirements**

The program should print the following menu to the console and allow the user to choose one of the options. The user will be able to create reports as often as they like. The program will terminate when exit is selected.

**Main Menu**

1. **Ship Reports**
2. **Occupation Report**
3. **Age Report**
4. **Exit**

**Enter Choice :**

1. **Ship Reports**

The user is given a list of ships which appear in the file and is asked to choose a ship.

The details of a ship, including the number of passengers, are written to the console.

This is followed by a list of all the passengers on that ship.

Example based on the sample file:

DISPATCH 187 06-26-1851 049 : leaving from DONEGAL Arrived : 26/06/1851 with 49 passengers

First Name HANNAH : Last Name DONNELL

First Name JAMES : Last Name DORRISON

First Name MARGT. : Last Name GRAHAM

First Name JAMES : Last Name WARD

First Name EDWARD : Last Name CARNEY

First Name JOHN : Last Name CANADA

(…. and another 43 names)

1. **Occupation Report**

The number of passengers in each occupation across all the ships are written to the console.

1. **Age Report**

The number of passengers in each of the following age range across all the ships are written to the console. The age ranges are:

Infants : <1 year

Children: 1-12

Teenage: 12-19

Young adult: 20-29

Adult: 30+

Older Adult: 50+

Unknown

[See notes on conversion/validation of the age field in the next section below]

# **3. Data Validation and Exception Handling**

1. Exception handling should be used for the file handling, if the file cannot be found or cannot be opened the program should write a message to the console and exit gracefully.
2. When reading in records from a file data validation should be carried out as follows:

If a line does not contain 10 comma separated values it can be discarded and a message should be written to the console. The rest of the lines should continue to be read.

If a date does not appear in the format you expect, the line it is on can be discarded and a message should be written to the console.

1. **Important Note:** Validating Dates

The file contains dates in the American Format i.e. MM/DD/YYYY as we are using a system date format of DD/MM/YYYY we have to give it the American context when parsing dates. The following code will try to parse string in the American format.

(DateOnly.TryParse(dateString,

CultureInfo.GetCultureInfo("us-EN"),

DateTimeStyles.None, out date ))

(ii) Important Note : Validating and converting ages.

Ages need to be converted to a numeric value for the age report. Any age that has a code of “Unknown”, or any invalidly formatted age can be converted to a value of –1 for reporting purposes.

A **valid age** can be in a text string with any of the following three formats (for example):

(i) Age: 24

(ii) Infant in months: 10

(iii) “Unknown”

# **4. Reflection on Assignment (Word document)**

Students must also submit a short reflection as a word document which follows the given template.

# **5. Coding Guidelines**

* Follow naming conventions using camelCase for private class , PascalCase for class and method names and public class Properties
* Write clean and well laid out code.
* Make sure you validate data and catch exceptions where appropriate ( e.g. opening files).
* Use /// comments on methods and classes.

**6. References**

The passenger file was adapted from a file extracted and downloaded from the US government archives. Due to size restrictions, a section of the file covering those embarking the ships in Donegal was downloaded.

**File unit:** Famine Irish Passenger Record Data File (FIPAS), 1/12/1846 - 12/31/1851  
**in the Series:** Records for Passengers Who Arrived at the Port of New York During the Irish Famine

<https://aad.archives.gov/aad/fielded-search.jsp?dt=180&cat=GP44&tf=F&bc=,sl>

**7. Hints on Approach**

**Start small and get one bit to work at a time.**

1. Get the outline menu working (you can re-use existng code here).
2. Read in data from files.
3. Check correct number of fields and correct date
4. Create classes and objects
5. Create reports – one at a time.