**PROG1780 – Programming Fundamentals: Fall 2020**

**Assignment 4**

**[Maximum points: 35]**

In this part of the assignment, you are required to create a C# Console Application project. The project name should be A4<LastName><FirstName>P1. For example a student with Last name Smith and first name John would name the project A4SmithJohnP1.

**~~This assignment is focused on you demonstrating the use of arrays~~**

~~Create a C# console program that creates and records information for a T-Shirt store. To manage the size of inventory, the owners only want to store data for a total of 12 T-Shirts. Assume the store can only hold three T-Shirt brands with one of four sizes for each brand which are Small (1), Medium (2), Large (3), XLarge (4).~~

**~~When the program is first run get the name of each of the three brands and store in an array called “brand+<yourLastName>” for example brandBrown. Use a for loop to perform this operation.~~**

~~the following information must be displayed in a menu:~~

* 1. ~~Add new T-Shirt details -Brand Name & Size (for example Fendi-4)~~
  2. ~~Edit existing T-Shirt details (Brand Name & Size)~~
  3. ~~Display all T-Shirts in store (display the Brand Name & Size)~~
  4. ~~Delete T-Shirt Information~~
  5. ~~Exit the program~~

~~After a choice from the main menu is selected, the program accurately performs the necessary operations and returns to the main menu unless option "E" is selected.~~

**Note the following for each option**

**A – Add new T-Shirt details**

* ~~For option "A" request the Brand Name and the Size using 2 separate requests~~.
  + ~~The Brand Name –must be one of the three brand names entered at the start of the program~~
  + ~~The Sizes - must be either number 1,2,3 or 4~~
* ~~put the final format together which must be Brand-Size example “Fendi-1”.~~
* ~~The program must display “Error T-Shirt record already exists" if one was already entered and previously stored (so this must be checked for before attempting to save~~.
* ~~You must use error exception handling for full marks~~ 
  + ~~throw an exception error if the user enters a number less than 1 or greater than 4.~~
* ~~In case of an invalid input, the user must be prompted to re-enter the value until a valid value has been entered.~~
* ~~If the input is correct, save the brand information to the array by searching the array, for the first available record spot (which can be a previously deleted brand (“NONE” – see option D below) OR a new record – once there are less than 20 T-Shirts stored.~~
* ~~notify the user that the record has been saved and repeat the process for option A – user should be able to enter new brand information until they enter “DONE” for the brand name field or when the array has 20 record. Use a while loop to perform this task.~~

**B ~~- Edit existing T-Shirt details (Brand Name & Size)~~**

* ~~For option "B" ask the user for the brand entry they need to edit; it must be entered in the format Brand-size (example Fendi-2)~~
* ~~If no record exists, the program must display " Brand record not found for that entry"~~
* ~~If the record exists, display the message “brand information found” then~~
* ~~Prompt the user to enter new values (“please enter updated information”)~~
  + ~~For each value as with menu option A, use separate requests for Brand and size and perform the same validations.~~
* ~~If the information entered is correct update that specific position in the array (that contained the old record) then notify the user that the record is updated – pressing enter should then display the main menu.~~

**~~C - Display all T-Shirts in store (display the Brand Name & Size)~~**

* ~~For option "C" the program must displays all the saved brand information to the screen.~~
* ~~After which when the user presses enter the program must return to the main menu – prompt the user to press enter to return to the main menu.~~

**~~D - Delete existing T-Shirt details (Brand Name & Size)~~**

* ~~For option "D" ask the user for the brand entry they need to delete; it must be entered in the format Brand-size (example Fendi-2)~~
* ~~If no record exists, the program must display " Brand record not found for that entry" then return to the main menu~~
* ~~If the record exists, display the message “brand information found are you sure you want to delete Y/N”~~ 
  + ~~If the user enters “Y” replace the brand information record with the value “NONE” then return to the main menu.~~
  + ~~IF the user enters “N” return to the main menu.~~

**Sample Test Plan - 5 marks:**

~~Please pay attention to the marking Rubric (available on class portal in same location as assignment - eConestoga) each of the~~ **~~items #2-10 under option A~~** ~~- represents a test item on your plan and must be represented as shown in the test plan example below:~~

Outline the test plan in this example format shown below.

Test Plan for Assignment #4 Coder: <your name > Tester: <your name > (in this case)

|  |  |  |
| --- | --- | --- |
| Test description From Rubric | Expected Result | Test Pass or Fail |
| <description of what is being tested> | <what you expect to happen when that feature is tested or action happens > | <If the expected result happens pass if it doesn’t happen fail> |
| **Example: User Notified when record is saved** | **Prompt shown on screen “Record Saved Successfully”** | **Pass** |

Adhere to coding and assignment submission standards/guidelines. Marks will be taken off, if these standards/guidelines are not followed. The format for submitting the assignment is as follows:

1. **eConestoga Submission:**

The **ENTIRE** project folder uploaded to the assignment folder under “Assignment 4”.

1. **Demonstration of program in class:**   
   Please have your program uploaded to eConestoga and running when you are ready to demonstrate your work.