Document Details			
Version Number	**Last Updated**	**Developed/Edited By**	**Validation Date**
1.3	**17/7/2019**	**Godwin/Robson**	**November 2018**

Training Package	ICT Information and Communications Technology Training Package
Qualification Title	ICT41015 AVZ7 Certificate IV in Computer Systems Technologies
	ICT40115 AWB5 Certificate IV in Information Technology
Course Title	Programming Fundamentals
Assessment Title	AT3 - Project

Brief Description of Assessment Task

Create an application that will allow a person to keep track of their Computer Games collection

Units of competency, elements to be assessed

National Code	SIN	Competency Title	Elements of Competency
ICTPRG414	AUV79	Apply introductory programming skills in another language	1.Apply basic language syntax and layout 2.Code using data structures 3.Code using standard algorithms 4.Debug code 5.Document activities 6.Test code
ICTPRG405	AUV52	Automate processes	1.Develop algorithms to represent solution to a given problem 2.Describe structures of algorithms 3.Design and write script or code 4.Verify and review script or code 5.Document script or code

Date of Assessment	Session 13	**Completed by**	Session 19
Instructions to Students	The analysis, design, coding, testing and simple documentation of a C#.NET application as described on the following page.		
Resources Required	Reference books / Internet / Blackboard / Visual Studio 2017/ MS Project		
Instructions to Lecturer/Assessor	Following the student demonstration collect and assess all project activities at the end of the session.		

Lecturer's Details (Add your lecturers details below)

Name	
Email	
Campus	

Students to sign this document when submitting an assessment

Date Submitted:	

STUDENT DECLARATION

- I have read and understand the details of the assessment.
- I have been informed of the conditions of the assessment and the appeals process.
- I agree to participate in this assessment.
- I certify that the attached is my own work.

Student ID	Student Name	Student Signature

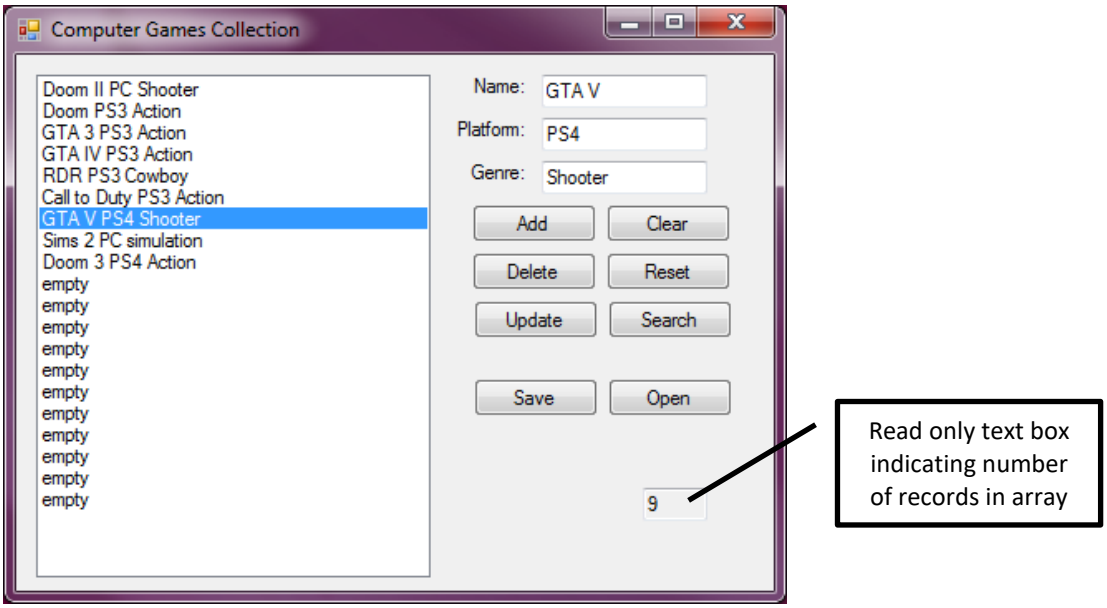
Assessment Feedback (Lecturer and Student Copy)			
Assessment Title	AT3 - Project		
Candidate name		**Attempt No**	
Assessor name			
Performance demonstrated by this assessment is	Satisfactory ❑	Not Yet Satisfactory ❑	
	Assessment outcome and feedback received on	Date	
Assessor Comments:			
Candidate signature: *(once feedback has been received)*		**Date**	
Assessor signature: *(once feedback has been provided)*		**Date**	

AT3 - Project

In this assessment you will create a new Windows Forms Application which will allow a person to keep track of their Computer Games collection. The purpose of this assessment is to demonstrate the use of multiple arrays to store "string" data. Therefore, you will be required to create three (3) global arrays of size 20 to store the three items of data; Name, Platform and Genre. The information implied by the following graphic needs to be read into three (3) string arrays of size 20 from a single Binary file when the program opens. The details of the project are as follows;

GUI Layout

The following GUI form layout is a recommendation and should be used as a starting point for your work. This example does not show all the buttons required for the final submission (ref Program Criteria). You will be required to add additional buttons and features to suit your specific interpretation of the application and program criteria. Add comments to each method and global variables that identifies the Program Criteria and explains the purpose of the code block. Debug and test your code to ensure full functionality before you demonstrate the application to your lecturer/assessor.



Program Criteria

Your final program will need to satisfy the following minimum criteria;

1. The program loads the computer games information from a single Binary called "mygames.dat" when the program starts into each of the three (3) Arrays. All the data saved and stored can be of type string. (An optional OPEN button can be added to select a specific file).
2. To add a new computer game; type the data values into the three text box fields and click the ADD button. The text fields should then be cleared (using a call to a custom clear method) once the record

has been added. All data added to the Array must be “real”, you cannot demonstrate or submit using data like: Name: “aa”, Platform: “bb”, etc.

3. To delete a record, select a record from the ListBox and click the DELETE button. The text box fields should then be cleared and the record removed from each of the three (3) arrays. The empty field should be removed; i.e. you can shuffle the data down the array (Hint, this could be done by re-sorting the arrays).

4. To edit the details of a game, select a game from the ListBox, then change the values in the input text box fields and click the UPDATE button. The text fields should then be cleared and the updated data displayed in the Listbox.

5. Add a RESET button to clear all the data records from the Arrays and initialize the Name field to “empty”. The ListBox and input text fields should also be cleared (Hint the tilde character could be used to preface the Name filed entry: this can also be used in the Bubble Sort).

6. Create a method to do the following: when a computer game Name is selected from the ListBox on the left, the details are displayed in the text box fields on the right.

7. All the game details should be displayed into the ListBox which is sorted alphabetically (using computer game Name field, ascending). The sort method MUST use the Bubble Sort algorithm, an optional SORT button can be added, you cannot use any built-in methods for this criteria. The computer game Name field should be unique as the collection might have several computer games each with the same platform type and genre. Hint: When the tilde character is used to preface the empty name the following IF statement in the Bubble Sort can be utilized: “if (string.CompareOrdinal(name[pos], name[pos - 1]) >= 0)”.

8. To find a computer game the user will type the Name into the text box field and click the BINARY SEARCH button. If the computer game is found the remaining text box fields should be populated. If the computer game is not found, then a message should be displayed and the text box fields cleared. The search algorithm must use a Binary Search algorithm, you cannot use any **built-in** methods for this criteria, ie do <u>not</u> use Array.BinarySearch().

9. Add a second search button that implements a linear search algorithm. To find a game the user will type the Name into the text box field and click the LINEAR SEARCH button. If the computer game is not found, then a message should be displayed and the text box fields cleared.

10. All data from the three (3) arrays should be written back to a single binary file called “mygames.dat” file when the program closes.

11. The SAVE button will allow the user to save all the data back to the “mygames.dat” file.

12. Each of the text box controls should have a tool tip text attached.

13. You can add additional features to enhance the UX experience. (for example, a message box to confirm the data has been save to file successfully)

Program Report

Create a formal report with 4 sections. Ensure your report has a title and content page; the footer will require page numbers and student details (name and ID). Your report will require the following headings;

Introduction

A full and detailed explanation of what the program is required to do.

Analysis

A statement of what data items need to be inputted, what processes need to be performed and what output is required. This will be a detailed explanation of each of the methods in your form. (use the example table below and fill with information pertaining to your computer games program)

Input:	*List the data inputs into the program, how does data get into your program?*
Process:	*List all the methods used by the program, what methods process the data?*
Output:	*List all the outputs from your program, how is data displayed and saved?*

Test Data

Create a Test Table of all the various UX tests that you will perform. These tests will ensure full functionality of your program and the various error trapping/messages used during normal usage. Your tests should include; add new data, edit data, delete data, search and save. All pathways of your code will need to be tested. Include screenshots of the program functioning using the test data.

You could use the following Test Table example as a starting point…

Test Case	Description	Expected outcome	Evidence
Case 1.	Add new game	Data added to Arrays, then displayed in the List Box	Ref Screen Capture 1.
Case 2.	Add new game (missing data	Error message	Ref Screen Capture 2.
Case 3.			
Case 4			
…			
…			
…			

Program Code

Include the method signatures (header) and comments of your C# code for each of the methods. Do not include the code inside the method. Ensure you add the global variables at the top of your program.

User Guide

Create a two page user guide and save in pdf format. The user guide should explain the usage of the application and each of the controls. Ensure your guide includes images and callout shapes that explain the function of all the buttons, etc.

Demonstration

You will be required to demonstrate the fully functioning program to your lecturer before you submit all the associated code and documents.

Submission Requirements

The following is a checklist to help you determine whether you have completed all requirements before you submit.

Project Submission Checklist	
Does your program satisfy all the Program Criteria?	YES ☐ NO☐
Have you completed the Report?	YES ☐ NO☐
Have you completed the User Guide?	YES ☐ NO☐
Have you demonstrated you working program to the lecturer?	YES ☐ NO☐

For this assessment, zip up the project folder and associated word documents, then upload

End of AT3 - Project