



Document Details			
Version Number	Last Updated	Developed/Edited By	Validation Date
003	30/11/2020	Stewart Godwin	Dec 2017
Training Package	ICT Information and Communications Technology Training Package Version 5.0 (Release 1)		
Qualification Title	ICT40518 BEH5 Certificate IV in Programming		
Qualification Title	ICA50718 BEG8 Diploma of Software Development		
Course/Cluster Title	Programming II		
Assessment Title	Project Assessment Task Two (AT2) Version B		
Brief Description of Assessment Task			
The Analysis, design, coding, testing and documentation of a program implementing multiple classes and file I/O as specified in the assessment tool description.			
Units of competency, elements to be assessed			
National Code	SIN	Competency Title	Elements of Competency
ICTICT403	AUU46	Apply software development methodologies	1. Determine and select appropriate methodology for a given activity 2. Apply the selected development methodology 3. Adjust project to suit appropriate methodology
ICTPRG418	AUV62	Apply intermediate programming skills in another language	1. Code using user defined data structures 2. Code using standard algorithms 3. Debug code 4. Document activities 5. Test code 6. Create an application
Date of Assessment	Session 8		Completed by Session 10
Instructions to Students	The analysis, design, coding, testing and project documentation of a C#.NET application as described on the following page.		
Resources Required	Reference books / Internet / Blackboard / Visual Studio 2017/ MS Project / Project Libre		
Instructions to Lecturer/Assessor	Assess each project using the exemplars. Ensure that each student gives a demonstration of their fully functional program. Use the attached Checklist to confirm functionality of student project.		
Lecturer's Details (Add your lecturers details below)			
Name			
Email			
Campus			
Students to sign this document when submitting an assessment			
Date Submitted:			
STUDENT DECLARATION			
<ul style="list-style-type: none"> <li>I have read and understand the details of the assessment.</li> <li>I have been informed of the conditions of the assessment and the appeals process.</li> <li>I agree to participate in this assessment.</li> <li>I certify that the attached is my own work.</li> </ul>			
Student ID	Student Name	Student Signature	



Assessment Feedback (Lecturer and Student Copy)			
Assessment Title	Project Assessment Task Two (AT2) Version B		
Candidate name		Attempt No	
Assessor name			
Performance demonstrated by this assessment is	Satisfactory <input type="checkbox"/>	Not Yet Satisfactory <input type="checkbox"/>	
	Assessment outcome and feedback received on	Date	
Assessor Marking Guide and Comments:			
Criteria	s/ns	Comment	
Program reads data from 3 binary files and writes data into each Array			
Add Drone button functionality; checks textboxes for data checks for book data type (Price – Double) add product to Array has error message for incorrect input			
Add Customer functionality; checks textboxes for data has error message for incorrect input add unknown customer add customer to Array			
Add Transaction functionality checks textboxes for data has error message for incorrect input add transaction to 2D array			
Listbox selection populates corresponding textboxes Correct functionality for “Drones” Correct functionality for “Customers” Correct functionality for “Transactions” Correct functionality for populating Transactions, Customer and Drones			
Textbox Double Click in CustomerID and SerialNumber clears all the associated input text boxes.			
All Drone data is sorted using a simple sort algorithm after each new drone is added.			



Create a simple linear search for Customer ID. Add a context menu strip for this search method. User must enter value and right click CustomerID textbox to launch search method.		
Program reads data from each Array and writes data to 3 binary files		
All data is realistic (must represent real names, and drone data)		
All major textboxes must have tooltips		
Report must have; Introduction Analysis with diagram Project plan with Gantt chart Activity diagrams for the binary file handling Test data/table and screen captures Code Method Signatures (headers) and comments		
Training guide with diagrams and explanations.		
<b>Candidate signature:</b> <i>(once feedback has been received)</i>		<b>Date</b>
<b>Assessor signature:</b> <i>(once feedback has been provided)</i>		<b>Date</b>

## Assessment Instructions

### Case Study – Ace Drones

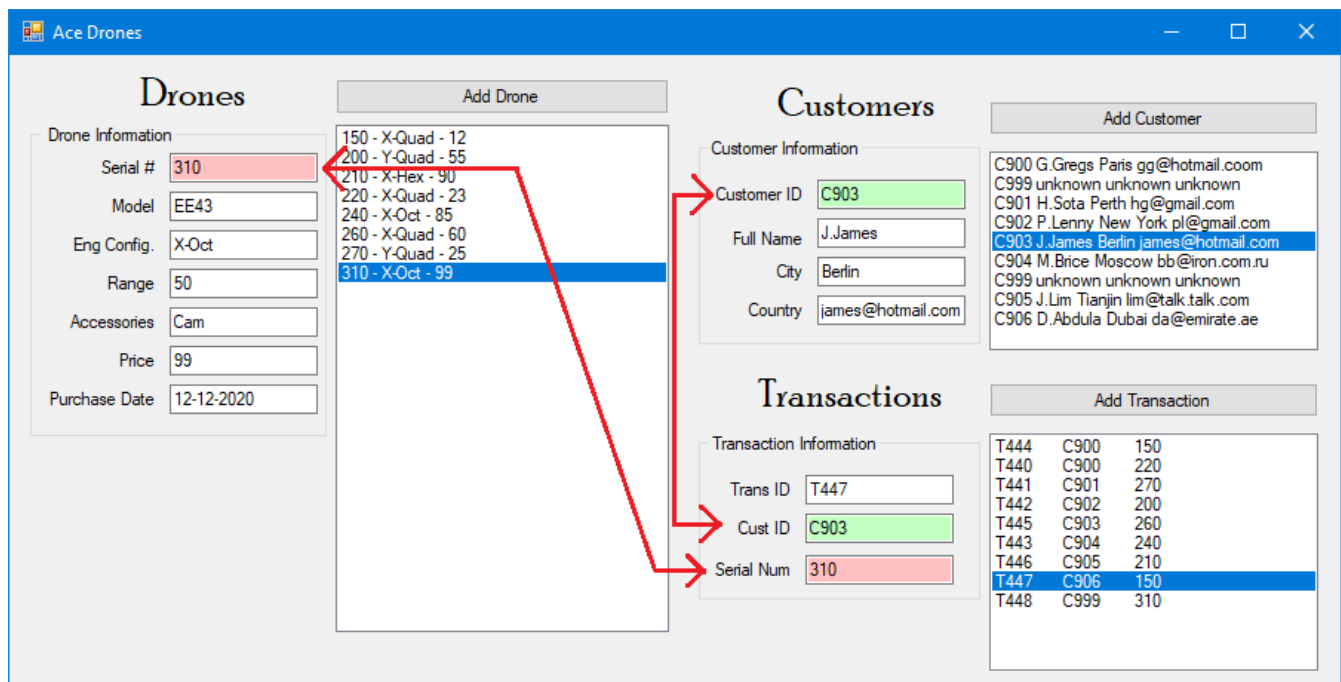
Ace Drones is a company building and selling military grade security drones around the world, they have a team of 15 sales consultants who visit various government agencies and private organisations. They require a simple application to maintain a list of the drones they have sold.

- A Drone class which has the following attributes; **serialNumber**, **model**, **engineConfiguration**, **range**, **accessories**, **price** and **purchaseDate**, these must be private, with associated assessor methods to store and retrieve information.
- A Customer class to store basic information about the customer; create a class with private attributes and associated assessor methods to store and retrieve information,
  - **customerID**, **name**, **city**, **country**.
  - If no details are provided, they can be grouped as a single unknown customer with an ID of "C999". The other fields can be filled with "unknown".
- Transactions do not have a class; this data will be stored in a 2D array which contains the following info
  - **transactionNumber**, **customerID**, **serialNumber**.

In the role of Senior Programmer for CITE Managed Services you have completed the research and analysis of the Ace Drone Application. Your final task is to create a fully functional application to be demonstrated with real data. The final submission will include the application and a detailed report. Complete the following questions to satisfy the requires of this assessment;

### Question 1. User Interface Design

The Graphical Interface is as follows



The screenshot shows the 'Ace Drones' application window. It is divided into three main sections: Drones, Customers, and Transactions.

**Drones Section:** Includes a 'Drone Information' form with fields for Serial # (310), Model (EE43), Eng Config. (X-Oct), Range (50), Accessories (Cam), Price (99), and Purchase Date (12-12-2020). Below the form is a table of drone data:

Serial #	Model	Eng Config.	Range	Accessories	Price	Purchase Date
150	X-Quad	12				
200	Y-Quad	55				
210	X-Hex	90				
220	X-Quad	23				
240	X-Oct	85				
260	X-Quad	60				
270	Y-Quad	25				
310	X-Oct	99				

**Customers Section:** Includes a 'Customer Information' form with fields for Customer ID (C903), Full Name (J.James), City (Berlin), and Country (james@hotmail.com). Below the form is a table of customer data:

Customer ID	Full Name	City	Country
C900	G.Gregs Paris	gg@hotmail.com	
C999	unknown	unknown	unknown
C901	H.Sota Perth	hg@gmail.com	
C902	P.Lenny New York	pl@gmail.com	
C903	J.James Berlin	james@hotmail.com	
C904	M.Brice Moscow	bb@iron.com.ru	
C999	unknown	unknown	unknown
C905	J.Lim Tianjin	lim@talk.talk.com	
C906	D.Abdula Dubai	da@emirate.ae	

**Transactions Section:** Includes a 'Transaction Information' form with fields for Trans ID (T447), Cust ID (C903), and Serial Num (310). Below the form is a table of transaction data:

Trans ID	Cust ID	Serial Num
T444	C900	150
T440	C900	220
T441	C901	270
T442	C902	200
T445	C903	260
T443	C904	240
T446	C905	210
T447	C906	150
T448	C999	310

Red arrows indicate data flow: from the 'Serial #' field in the Drone form to the 'Serial Num' field in the Transaction form, and from the 'Customer ID' field in the Customer form to the 'Cust ID' field in the Transaction form.

Figure 1



## Question 2. Program Functionality

The program functionality must satisfy the following criteria:

1. The program loads the drone information from a binary file called **drones.dat** when the program starts into an appropriate single array structure. Array size of 20.
2. The program loads the customer information from a binary file called **customers.dat** when the program starts into an appropriate single array structure. Array size of 20.
3. The program loads the transaction information from a binary file called **transactions.dat** when the program starts into an appropriate 2D array structure.
4. When the Drone ADD button is clicked,
  - a The information in the textboxes is checked to verify the data type,
  - b If information is in too many or too few input boxes then an error message is generated,
  - c If the information is correct then a Drone object is created and the object is added to the array structure and the **serialNumber**, **engineConfiguration** and **price** are displayed in the listbox. Add hyphens between data items.
  - d The array is sorted by **serialNumber** in ascending order. Use a simple Bubble sort algorithm, when a record is swapped ensure the index/object is passed by reference to a separate swap method.
  - e Once the new drone has been added to the array and displayed in the listbox the input text boxes are cleared.
5. When the Customer ADD button is clicked,
  - a The information in the textboxes is checked to verify it is present,
  - b If information is not present then a dialog popup is generated with a request to generate a default customer using a yes/no option,
  - c If the user selects YES then a default customer is added (refer Case Study info and Figure 1.)
  - d If the user selects NO then no further action is taken,
  - e When all the correct information is present a Customer object is created and the object is added to the array structure and all the data is displayed in the listbox.
  - f Once the new customer has been added to the array and displayed in the listbox the input text boxes are cleared.
6. When a record in either of the two upper listboxes (Drone or Customer) is clicked, the information relating to that record is to be added to the correct Drone and Customer textboxes. The appropriate information is also added to the textboxes under Transaction, (refer: red arrows in Figure 1).
7. Before the Transaction ADD button is clicked the user must first select a customer and drone. This action will populate the **customerID** and **serialNumber** textboxes in the transactions groupbox (ensure these textboxes are read only).
  - a The information in the input textboxes is checked to verify all data is present, otherwise generate an error message,
  - b If all data is present a Transaction object is created and the object is added to the 2D array structure and the information is added to the listbox. Use a tab delimiter between the data items in the listbox.

- c When a record in the transaction listbox is clicked the transaction information is filled into the remaining transaction input text boxes, and the appropriate Book and Customer records are selected in the upper listboxes. This should autofill the upper textboxes.
8. If the user double clicks the **serialNumber** input textbox it will clear the all the input textboxes associated with the Book and allow a new book to be entered.
9. If the user double clicks the **customerID** input textbox it will clear the all the input textboxes associated with the Customer and allow a new customer to be entered.
10. To search for a customer the user will enter text into the **customerID** input textbox. The user then right clicks the **customerID** input textbox and a context menu will appear with a search button. If the record is found the other fields will be populated. If the record is not found generate an error message box. The search algorithm must use the simple built-in binary search.

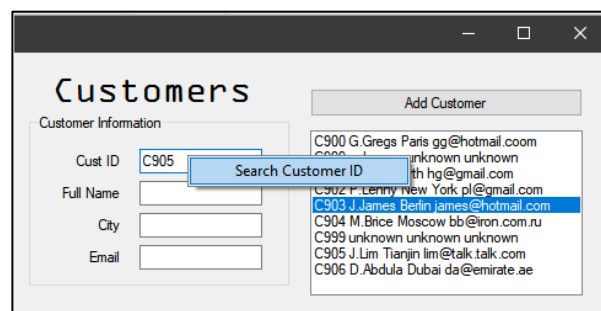


Figure 2

11. If the user double clicks the **transactionID** input textbox it will clear the all the input textboxes associated with the Transaction and allow a new transaction to be entered.
12. All data should be written back to the three binary files when the form closes.
13. Add code comments to all Methods and Classes. Ensure all key aspects of your code are fully documented (do not use complex or technical terms)
14. Add realistic data to test the application
  - o 8 customers (max of 3 unknown),
  - o 15 drones,
  - o 10 transactions
15. Each of the major controls should have a tool tip text attached, refer Figure 3.

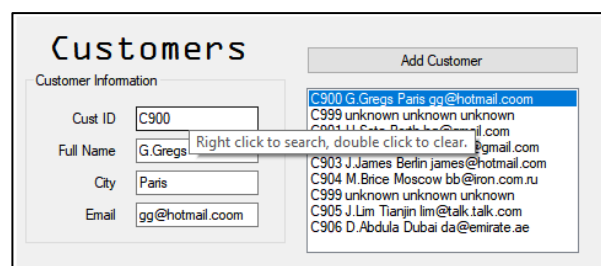


Figure 3



### Question 3. Report

Create a Formal report with the following sections. Saved as Docx format (or similar). 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 detailed explanation of what the program is required to do,
  - What development methodology did you use for this project, explain why you believe that it is the best in this situation (include a diagram)
- Analysis
  - A statement and explanation for each of the following;
  - What data needs to be inputted into the program,
  - What processes need to be performed while the program is running,
  - What output is generated when the program closes?
- Project Plan
  - A statement and explanation of the following;
  - List the tasks that needed to be performed to create this project
  - List the job title for each task, when and in what order (UI designer, programmer, etc)
  - List the physical resources required to complete the project (PC, software, etc)
  - Draw a Gantt chart to show the above information (Project file)
  - NOTE: this must be accomplished using MS Project (or similar Project Libre)
- Algorithm Design
  - A UML diagram or Pseudo Code for the following major code methods,
    - Drone ADD button
    - Drone Load binary file
    - Drone Save binary file
    - Sort algorithm
- Test Data and Evidence
  - Describe the test data and activities that will be used to test the design and program functionality. Include a test table of the data to be used and the expected outcome. Ensure all the code is tested.
  - Run a series of tests using the test data previously stated. Record your tests and collect screenshots of the program functionality.
- Internal Documentation (Code Comments)
  - List all code comments for all methods and global variables (Method Signature and comment)
- Recommendations and Improvements
  - Describe all concerns and improvements that can be added to you program.
  - Explain the limitations, error trapping, data filters and usability/portability of your application.

Ensure that you include all references to all resources you used to complete your responses.



#### Question 4. Demonstration and Observation

Before you can submit your final project and associated files you will need to demonstrate the fully functional program.

The following observation checklist will be used to verify your program

Observation Checklist			
Criteria		Yes	No
#1, 2, 3	Loads "drones.dat", "customers.dat" and "transactions.dat" from binary file	<input type="checkbox"/>	<input type="checkbox"/>
#4	Correct functionality for Add Button in "Drones"	<input type="checkbox"/>	<input type="checkbox"/>
#5	Correct functionality for Add Button in "Customers"	<input type="checkbox"/>	<input type="checkbox"/>
#7	Correct functionality for Add Button in "Transactions"	<input type="checkbox"/>	<input type="checkbox"/>
#6, 11	Correct functionality for populating Transactions	<input type="checkbox"/>	<input type="checkbox"/>
#7.c	Correct functionality for drones listbox	<input type="checkbox"/>	<input type="checkbox"/>
#6	Correct functionality for customer listbox	<input type="checkbox"/>	<input type="checkbox"/>
#6	Correct functionality for transaction listbox	<input type="checkbox"/>	<input type="checkbox"/>
#8, 9	Double click method to clear textboxes	<input type="checkbox"/>	<input type="checkbox"/>
#10	Context menu for Search	<input type="checkbox"/>	<input type="checkbox"/>
#12	Save method to write data to binary files	<input type="checkbox"/>	<input type="checkbox"/>
#15	Tool tips on all controls	<input type="checkbox"/>	<input type="checkbox"/>
#13	Fully commented methods and classes	<input type="checkbox"/>	<input type="checkbox"/>
#14	Demonstration of working program	<input type="checkbox"/>	<input type="checkbox"/>
Final outcome:			

#### Submission Requirements

Following is a checklist to help you check whether you have completed all requirements. Submit each as a separate file

Files	Checklist
Zipped solution folder	<input type="checkbox"/>
Report	<input type="checkbox"/>

End of Assessment Task Two (AT2)