|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 7.2) | | |
| **Qualification National Code and Title** | ICT40120 Certificate IV in Information Technology (Release 3) | **State code** | BFF9 |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 1) | **State code** | BGJ4 |
| **Assessment Title** *(as per DAP)* | Assessment Task One (Individual Project) | | |
| **Unit National Code & Title** | ICTPRG443 Apply intermediate programming skills in different languages | | |
| ICTICT430 Apply software development methodologies | | |
| ICTICT449 Use version control systems in development environments | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date Due** | Week Ten | | **Date Received** | | 01/02/2022 | |
| **Student Name** | Poorav Sharma | | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work: | | | | | |
| **Assessor Name** |  | | | | | |
| **Assessment Decision** | Satisfactory | | | Not Yet Satisfactory | | |
| **Assessor Signature** |  | | | **Date** | |  |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | | **Re-assessment Date:** | | Week Twenty |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
| *Via Blackboard (LMS) – Please check [Grade] section.* | | | |
| **Feedback from student** | | | |
| *Via Blackboard (LMS) – Please use [Comment] section during submission.* | | | |
| **Student signature** |  | **Date** |  |

|  |  |
| --- | --- |
| **Assessment Instructions** | |
| **TO THE ASSESSOR** |  |
| Type of Assessment | Individual Portfolio |
| Duration of the assessment | 8 class sessions (Weeks 2-10) |
| Location of assessment | Classroom |
| Conditions | Assessor to ensure that the noise levels, natural interactions and time variances are maintained as it would be in the Software Development industry.  Learners are required to complete the required tasks in class and submit the required documentation electronically via Blackboard |
| Elements and Criteria | As detailed in the assessment plan  You are required to make sure that all students meet the elements, performance criteria and oral communication items as outlined in the provided solution |
| **TO THE STUDENT** |  |
| Purpose of Assessment | You are required to show you can:  ICTPRG443 Apply intermediate programming skills in different languages   * Demonstrate your skills and knowledge by creating, coding, debugging and testing code * Establish user requirements and then research and collect information about data structures to provide suitable solutions. * Manage time and tasks to complete a series of coding and documentations problems   ICTICT430 Apply software development methodologies   * Select traditional and non-traditional systems development methodologies * Apply selected software methodology to project plan which identifies resources and control structures * Document analysis for approval to external stakeholders.   ICTICT449 Use version control systems in development environments   * Prepare and evaluate version control systems * Install and configure a version control system * Create and upload code to version control system * Test and review logs on version control system   The student must demonstrate the ability to complete the tasks outlined in this assessment and is expected to use systematic analytical processes and effect time management to meet the goals/deadlines outlined in the DAP. |

|  |  |
| --- | --- |
| Allowable Materials | Blackboard Topic One: SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with MS Visual Studio, MSOffice.  Internet Access to MSDN, GitHub and www.citems.com.au/ |
| Reasonable Adjustment | In some circumstances, adjustments to assessments may be made for you. If you require support for literacy and numeracy issues; support for hearing, sight or mobility issues; change to assessment times/venues; use of special or adaptive technology; considerations relating to age, gender and cultural beliefs; format of assessment materials; or presence of a scribe you need to inform your lecturer. |
| Assessment Submission | All questions and programming activities must be attempted. All written answers must be submitted in this assessment document in the appropriate space.  Use of research tools and peers in formulating answers are acceptable – but work submitted must be your own work.  Final project documentation is to be uploaded to the appropriate area in the Blackboard course created for this unit.  If you are marked as NYS (Not Yet Satisfactory) on your first attempt, you will be provided with another opportunity to re-attempt the assessment. |
| Portfolio Description | A project of programming tasks and written questions which should be completed in class and finished in the students’ own time on a weekly basis as per the Delivery and Assessment schedule.  Question 1 – Project Specifications  Question 2 – Version Control Specifications  Question 3 – UML Diagram  Question 4 – Development Methodologies  Question 5 – Methodology Analysis and Selection  Question 6 – Project Plan  Question 7 – Manage Version Control System  Question 8 – Prototype Development  Question 9 – Version Control Update  Question 10 – Data Structure Matrix  Question 11 – UML Activity Diagram  Question 12 – Debug Test Report  Question 13 – Post Development Analysis  Question 14 – Demonstration and Submission |

## Scenario

You have accepted the role of a Senior Programmer for CITE Managed Services, your task is to develop a fully functional wiki application for the junior programmers. In Computer Science there are many different categories and definitions for Data Structures, most of these terms are used in the CITE software development department, however, CITE management would like to see a uniform definition and cataloguing of this information. They have supplied some basic details but would like you to complete a feasibility study and create a working prototype application. A rudimentary interface design has been provided along with a list of proposed program criteria which the prototype application must include.

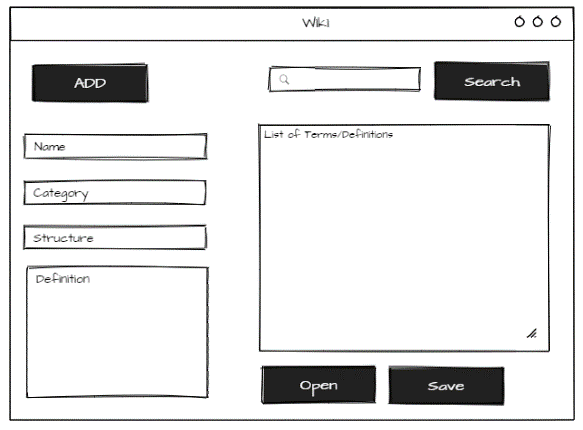
Before the project can move to the next stage CITE management would like a report on the full development process and related documentation. Review the proposed program criteria and answer the associated questions. Use the supplied template forms to present your answers. Finally develop a working prototype using Microsoft Visual Studio C# and GitHub version control. The purpose of the assessment is to demonstrate to CITE management how this project can be achieved. If you do not have a GitHub account you should sign up for the free version, this will be used again in other courses (https://github.com).

You should consult with the CITE representative (Your Lecturer) if you are unsure about any of the problems or questions. Your primary research should focus on the resources on the Blackboard and CITEMS website, additional information can be collected from the Internet, ensure all sources are referenced at the end of your submission. You must write your answers in the standard templates provided in this assessment task document.

## Proposed Program Criteria

|  |  |
| --- | --- |
| * The user can select an item from the list box and the corresponding information will be displayed in the four text boxes (on the Left), * The user can search for an item which will be displayed in the four text boxes, * Search input box must clear if search unsuccessful, * The user can add a new item, * All user interactions must have full error trapping and feedback messaging, * The wiki prototype will load and save data when the buttons are clicked, | * All wiki data is stored/retrieved using a binary file format, * A double mouse click in the search text box will clear the search input box, * The prototype must use a two-dimensional array of type string, * The Data Structure information must be defined using the following attributes: Data Structure Name, Category, Structure and Definition, (refer Data Structure Matrix in Question 10). |

## Proposed Interface Design



## Question 1

You are required to create a list of all the User Requirements for the wiki prototype application. Then list all the User Interactions and Specifications of the wiki application. Fill in the relevant sections of the following Project Specifications template to answer question one.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project Specifications | | | | | |
| Project Name | | | Wiki Prototype | | |
| Date | | | 01/02/2022 | | |
| Developer Name | | | Poorav Sharma | | |
| User Requirements | | | | | |
| Req. # | | Description | | Importance | Notes |
| 1 | | List box to display items to select | | Very important |  |
| 2 | | Four text boxes that display information of selected items | | Very important |  |
| 3 | | Search bar to search for specific items | | Important |  |
| 4 | | Add button to add new items | | Very important |  |
| 5 | | Full error trapping and feedback messaging | | Very important |  |
| 6 | | Save button to save updated list | | Very important | Save as binary file |
| 7 | | Load button to open saved list | | Very important |  |
| 8 | | Double click can clear search bar | | Least important |  |
| 9 | | Delete button to delete existing items from list | | Very important |  |
| 10 | | Add, Edit and delete information | |  |  |
|  | | *Add more lines as required* | |  |  |
| User Interaction and Specifications | | | | | |
| How will the application behave and what GUI specifications are required. | | | | | |
|  | User can select an item from the list box and the corresponding information will be displayed in the four text boxes | | | | |
|  | User can search for an item which will be displayed in the four text boxes | | | | |
|  | Search input box must clear if search unsuccessful | | | | |
|  | User can add, edit and delete items | | | | |
|  | User interactions must have full error trapping and feedback messaging | | | | |
|  | User can load and save data when the buttons are clicked | | | | |
|  | Double mouse-click in the search text box will clear the search input box | | | | |
| *Add more lines as required* | | | | | |

# Question 2

CITE currently use GitHub as their primary source control; however, they would like you to investigate/research an alternative to GitHub. The purpose is to ensure CITE have chosen the best version control system for software development. Fill in the relevant sections of the following Version Control Specifications template to answer question two.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version Control Specifications | | | | |
| GitHub VCS | | | Alternative VCS | |
| VCS Name: | | GitHub | VCS Name: | CVS |
| URL: | | <http://github.com> | URL: | <https://savannah.nongnu.org/projects/cvs> |
| Major Features | | | | |
| List all the major features associated with each version control system | | | | |
| 1. | Provides strong support for non-linear development. | | Client-server repository model. | |
| 2. | Distributed repository model. | | Multiple developers might work on the same project parallelly. | |
| 3. | Compatible with existing systems and protocols like HTTP, FTP, ssh. | | CVS client will keep the working copy of the file up-to-date and requires manual intervention only when an edit conflict occurs | |
| 4. | Capable of efficiently handling small to large sized projects. | | Keeps a historical snapshot of the project. | |
| 5. | Cryptographic authentication of history. | | Anonymous read access. | |
| 6. | Pluggable merge strategies. | | ‘Update’ command to keep local copies up to date. | |
| 7. | Toolkit-based design. | | Can uphold different branches of a project. | |
| 8. | Periodic explicit object packing. | | Excludes symbolic links to avoid a security risk. | |
| 9. | Garbage accumulates until collected. | | Uses delta compression technique for efficient storage. | |
| *Add more lines as required* | | |  | |
| Recommendation: Which VCS would you choose and why? | | | | |
| *Add your detailed answer here…*  I would choose GitHub as my VCS because it has a very fast and efficient performance. It is cross platformed and the code changes can be easily and clearly tracked. It is easily maintainable and robust. It offers an amazing command line utility known as git bash and additionally it offers GIT GUI where you can quickly re-scan, state change, sign off, commit & push the code with just a few clicks. | | | | |

## Question 3

Create a UML Diagram for the 2D wiki data structure information. Ensure you have added the data structure and attributes. Fill in the relevant sections of the following UML Diagram template to answer question three.

|  |  |
| --- | --- |
| UML Diagram | |
| Attributes | |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| Data Structure | |
|  | |
| Diagram | |
|  | |

## Question 4

You are required to create a comparison of four (4) software development methodologies that would be suitable to create the wiki prototype application. Your comparisons must include both traditional and non-traditional system development methodologies. Complete the following Development Methodologies template to answer question four.

|  |
| --- |
| Development Methodologies |
| 1st Methodology Name \_\_Agile\_\_\_\_ |
| Description: Agile methodology is used to minimize risk when adding new functionality (such as bugs, cost overruns, and changing requirements). In this method teams develop the software in iterations that contain mini-increments of the new functionality. |
| Diagram: |
| Advantages:   * Allows software to be released in iterations. * Iterative releases allow teams to find and fix defects. |
| Disadvantages:   * Requires on real time communication so new users often lack documentation to get up to speed. * Requires a huge time commitment from users and developers must fully complete each feature with each iteration for user approval which can be labour intensive. |
| 2nd Methodology Name \_\_\_Waterfall\_\_\_\_ |
| Description: The waterfall method is the most traditional software development method. This method is a rigid linear model that consists of sequential phases (requirements, design, implementation, verification, maintenance) focusing on distinct goals. Each phase must be fully completed before the next phase can start. There’s usually no process for going back to modify the project or direction. |
| Diagram: |
| Advantages:   * The linear nature of the waterfall development method makes it easy to understand and manage. * Less experienced project managers and project teams benefit the most from using the waterfall development methodology. |
| Disadvantages:   * The waterfall development method is slow and costly due to its rigid structure and tight controls. |
| 3rd Methodology Name \_\_\_\_ DevOps\_\_\_\_ |
| Description: DevOps deployment centres on organizational change that enhances collaboration between the departments responsible for different segments of the development life cycle, such as development, quality assurance, and operations. |
| Diagram: |
| Advantages:   * Reduces time to market products * Improves customer satisfaction, product quality, and employee productivity and efficiency due constant development. |
| Disadvantages:   * Some customers don’t want to continuous update their system * Some companies have regulations that require extensive testing before a project can move to operations phase. * If different departments use different environments, undetected issues can slip into production * Some quality attributes require human interaction, which slows down the delivery |
| 4th Methodology Name \_\_\_ Rapid \_\_\_\_ |
| Description: Rapid application development (RAD) is a condensed development process that produces a high-quality system with low investment costs. RAD process allows the developers quickly adjust to shifting requirements in a fast-paced and constantly changing market.” The ability to quickly adjust is what allows such a low investment cost. |
| Diagram: |
| Advantages:   * Useful for small to medium projects that are time sensitive. * Effective for projects with a well-defined business objective and a clearly defined user group |
| Disadvantages:   * Requires a stable team composition with highly skilled developers and users who are deeply knowledgeable about the application area. Organizations that don’t meet these requirements don’t benefit from RAD |
| References |
| *Add references as required*  *Synopsys Editorial Team. (2019, July 5). Top 4 software development methodologies | Synopsys. Software Integrity Blog.* [*https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/*](https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/) |

## Question 5

Refer back to question four and answer these two questions:

* What selection criteria determined your choice of the four development methodologies? Create a list of your section criteria.
* What methodology from question four would you recommend for this project? List your reasons why this is your preferred option.

Complete the following Methodology Analysis and Selection template to answer this question.

|  |
| --- |
| Methodology Analysis |
| Selection Criteria |
| Easy to implement changes |
| Can add new fun functionality if required by the client |
|  |
| *Add more lines as required* |
| Methodology Selection |
| Methodology Name \_\_\_**Agile**\_\_\_\_\_ |
| Justification (reasons for selection) |
| Agile methodology minimize risk when adding new functionality (such as bugs, cost overruns, and changing requirements). It allows teams develop the software in iterations that contain mini-increments of the new functionality. The iterative development allows teams to find and fix defects before when adding new functionality. |
| *Add more lines as required* |

## Question 6

Using your recommended methodology from question five, create an initial project plan. List and describe all the tasks required to complete the development of the wiki application. Use the following Project Plan template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | Wiki Prototype | | |
| Date | 22/02/2022 | | |
| Developer Name | Poorav Sharma | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| Form design | Designing | Design the form |  |
| Code the program | Coding | Build the program |  |
| Debug the program | Debugging | Debug your program and fix errors |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| *Add more lines as required* |  |  |  |

## Question 7

CITE has authorised the usage of GitHub as the recommended Version Control System (VCS) because it is fully integrated into Visual Studio, create your own GitHub account (you can use your existing account) then download and install the desktop interface onto your local PC. Create a suitable folder structure for both the local and cloud systems to accommodate the development of the wiki prototype application. Complete the following Manage Version Control System template to answer this question.

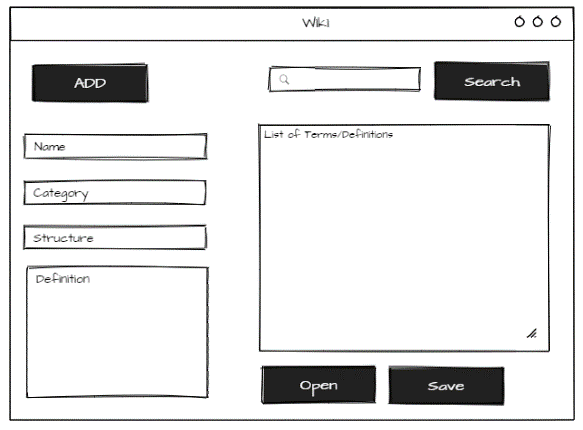
Consult with your lecturer if you wish to use an alternative source control service. Any alternative source control must support a local desktop installation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Manage Version Control System | | | | | | |
| VCS Name | |  | | | | |
| Version Number | |  | Date |  | | |
| Supported Platforms | |  | | | | |
| Installation Information | | | | | Yes | No |
| Has the cloud VCS account been created and named correctly? | | | | |  |  |
| Has the cloud VCS folder structure been created? | | | | |  |  |
| Has the desktop VCS software installed correctly? | | | | |  |  |
| Has the desktop VCS folder structure been created? | | | | |  |  |
| Are the two VCS resources linked? | | | | |  |  |
| Comments/Issues/Problems | | | | | | |
|  | | | | | | |
| Account Details Checklist | | | | | | |
| Repository Name: |  | | | | | |
| URL |  | | | | | |
| Desktop Screen Shot |  | | | | | |
| Cloud Screen Shot |  | | | | | |

## Question 8

Create the wiki prototype to demonstrate how a collection of information can be stored using a Windows Forms Application. This prototype application will utilise a two-dimensional array with 12 rows and 4 columns (**use a** **simple 2D string array – not collections).** Use the hardware and software supplied in the classroom to accomplish the development, debugging and internal documentation of the prototype. Use the Version Control System from Question 7 to manage your code during the development; ensure you record these commits as a series of screen shots to be included in Question 9 (ie start, working, final). Your code must adhere to the CITEMS software development standards. (refer http://www.citems.com.au/)

The following user interface is provided as a starting point for your prototype development.



**Note:** You are not permitted to use a class structure; this assessment is a demonstration of a 2D string array.

The following programming criteria and features are required;

### Programming Criteria

1. Create a global 2D string array, use static variables for the dimensions (row, column),
2. Create an ADD button that will store the information from the 4 text boxes into the 2D array,
3. Create a CLEAR method to clear the four text boxes so a new definition can be added,
4. Write the code for a Bubble Sort method to sort the 2D array by **Name** ascending, ensure you use a separate **swap** method that passes (by reference) the array element to be swapped (do not use any built-in array methods),
5. Write the code for a Binary Search for the **Name** in the 2D array and display the information in the other textboxes when found, add suitable feedback if the search in not successful and clear the search textbox (do not use any built-in array methods),
6. Create a display method that will show the following information in a List box: Name and Category,
7. Create a method so the user can select a definition (Name) from the Listbox and all the information is displayed in the appropriate Textboxes,
8. Create a SAVE button so the information from the 2D array can be written into a **binary file** called ***definitions.dat*** which is sorted by **Name,**
9. Create a LOAD button that will read the information from a binary file called ***definitions.dat*** into the 2D array,
10. All code is required to be adequately commented. Map the programming criteria and features to your code/methods by adding comments above the method signatures. Ensure your code is compliant with the CITEMS coding standards (refer http://www.citems.com.au/).

**Note:** The exact requirements of the Programming Criteria are essential. Any variation from them will need to be corrected in order to achieve a satisfactory performance.

## Question 9

At the conclusion of the code development record the Version Control changes, commits and pull requests with a series of suitable screen shots. Complete the following Version Control Update template to answer this question. A minimum of three screen captures is required (ie start, working, final)

|  |  |
| --- | --- |
| Version Control Update | |
| Repository Name: |  |
| URL |  |
| Desktop Screen Shots |  |
| Cloud Screen Shots |  |

## Question 10

Create test input data by researching and providing a definition for the 12 data structures listed in the Data Structure Matrix template below. The definitions must be between 20-40 words and contain real information which will be entered and saved by the wiki prototype. Add your definitions to the following Data Structure Matrix template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| Data Structure Matrix | | | |
| NAME | CATEGORY | STRUCTURE | DEFINITION |
| Array | Array | Linear |  |
| Two Dimension Array | Linear |  |
| List | List | Linear |  |
| Linked list | Linear |  |
| Self-Balance Tree | Tree | Non-Linear |  |
| Heap | Non-Linear |  |
| Binary Search Tree | Non-Linear |  |
| Graph | Graphs | Non-Linear |  |
| Set | Abstract | Non-Linear |  |
| Queue | Linear |  |
| Stack | Linear |  |
| Hash Table | Hash | Non-Linear |  |
| References | | | |
| *Add references as required* | | | |

## Question 11

Create a detailed UML Activity Diagram for the Binary Search method. Start by copying your C# code into the right side of the UML Activity Diagram section, then add your UML Activity Diagram in the left side. Now, using the code and the UML diagram identify breakpoints so all major pathways are tested. Update the C# Code on the right section to identify the breakpoints. The example is provided for clarification. Complete the following UML Activity Diagram template below.

|  |  |
| --- | --- |
| UML Activity Diagram | |
| Diagram | C# Code |
|  |  |

## Question 12

Using the breakpoints shown in the previous question as a starting point, utilise the debug features to debug, trace and test your Binary Search code. Ensure your code is error free and functions correctly (refer Programming Criteria Question 8.5) record and correct any errors. Your Debug Test Report must include appropriate evidence that your code functions as expected (references to screen captures). Complete the following Debug Test Report template below.

* Ensure you have entered 12 records from Question 10 before you begin testing,
* Place a break point at each Decision and Loop construct and record the data as it changes,
* Use a test data item that will be found (ie Stack),
* Use a test data item that will not be found (ie ArrayList).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Debug Test Report | | | | | | | | | | |
| Project Name | |  | | | | | | | | |
| **Method** | |  | | | | | | | | |
| **Description** | |  | | | | | | | | |
| **Level of Testing** | |  | | | | | | | | |
| **Developer** | |  | | **Tester** |  | | **Date** | |  | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
| 1 |  | |  | | |  | |  | |  |
| 2 |  | |  | | |  | |  | |  |
| 3 |  | |  | | |  | |  | |  |
| 4 |  | |  | | |  | |  | |  |
| 5 |  | |  | | |  | |  | |  |
| 6 |  | |  | | |  | |  | |  |
|  |  | |  | | |  | |  | |  |

## Question 13

Once you have completed coding and testing of this prototype application you can answer the following questions and complete the Post Development Analysis section in the template below.

1. What SDLC did you use during the development of the Prototype Application?
2. How effective was your project plan in developing the final prototype?
3. What alternative data structures could be used?
4. What constructive advice could you provide for the development of a similar project?

|  |
| --- |
| Post Development Analysis |
| Questions |
| What SDLC did you use during the development of the Prototype Application? |
| How effective was your project plan in developing the final prototype? |
| What alternative data structures could be used? |
| What constructive advice could you provide for the development of a similar project? |

## Question 14

### Demonstration and Submission

Demonstrate your working program to your lecturer using the realistic data from Question 10. Ensure your code is fully commented with your Name, ID, Date, Assessment Task placed above the workspace header. Ensure all the documentation has been completed and is ready for inspection, use the following Assessor Marking Guide to ensure all code and documentation is compliant.

### Assessor Marking Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Assessor Marking Guide | | Satisfactory | | Comment |
| **Questions** | | YES NO | |  |
| Q1 | Project Specifications: All proposed program criteria is listed within the User Requirements and User Interaction Specifications. |  |  |  |
| Q2 | UML Class Diagram: List four attributes and a simple UML class diagram showing Array |  |  |  |
| Q3 | Development Methodologies: List four different SDLC methodologies, No variations of a single methodology. |  |  |  |
| Q4 | Methodology Analysis: List several (three or more) selection criteria, Lists several (three or more) recommendations for SDLC |  |  |  |
| Q5 | Project Plan: Reflect selection from Question 4 and has all the SDLC tasks outlined |  |  |  |
| Q6 | Manage Version Control System: All fields are completed and there are screen shots of GitHub desktop and cloud |  |  |  |
| Q7 | Prototype Development: all program criteria and feature have been coded, software standard have been implemented in layout and comments |  |  |  |
| Q8 | Data Structure Matrix: All 12 data structure have a suitable definition. |  |  |  |
| Q9 | UML Activity Diagram: The code has breakpoints in the Decision and Iterative constructs, the activity diagram reflects the C# code. |  |  |  |
| Q10 | Testing and Debugging: All four bullet points have been tested and the results recorded. |  |  |  |
| Q11 | Post Development Analysis: All four questions have been answered. |  |  |  |
|  | **Assessment Decision**  Satisfactory  Not Yet Satisfactory | | | |

**Note:** All documentation must use the supplied templates/forms.

**Submit the zipped solution folder with relevant documents to Blackboard**

End of Assessment Task One