|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Release 8.1) | | |
| **Qualification National Code and Title** | ICT40120 Certificate IV in Information Technology (Release 4) | **State code** | BFF9 |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **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 Seven | | **Date Received** | | 20/7/2023 | |
| **Student Name** | James Chellew | | | | | |
| **Student Declaration** | I declare that the evidence submitted is my own work:  yes | | | | | |
| **Assessor Name** | Stewart Godwin | | | | | |
| **Assessment Decision** | Satisfactory | | | Not Yet Satisfactory | | |
| **Assessor Signature** | Stewart Godwin | | | **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** | **James Chellew** | **Date** |  |

|  |  |
| --- | --- |
| **Assessment Instructions** | |
| **TO THE ASSESSOR** |  |
| Type of Assessment | Individual Portfolio |
| Duration of the assessment | 7 class sessions (Weeks 1-7) |
| 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 Topics: 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 Class Details  Question 4 – Development Methodologies  Question 5 – Methodology Analysis and Selection  Question 6 – Manage Version Control System  Question 7 – Project Plan  Question 8 – Sign off and Approval  Question 9 – Prototype Development  Question 10 – Version Control Update  Question 11 – Data Structure Matrix  Question 12 – UML Activity Diagram  Question 13 – Debug Test Report  Question 14 – Post Development Analysis  Question 15 – 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.

## Client Program Criteria

The client was interviewed, and the following information was recorded. The client would like the end user to select a record from a display list and then have the corresponding information displayed in four text boxes on the left side of the interface. The end user should be able to search for an item which will be displayed in the four text boxes; after the search the search input box must clear and retains focus (cursor is inside the search text box) allowing the user to search for a new term.

The client requires that the end user has the option to add/edit/delete **any** of the four fields associated with an individual data structure record, the four fields are: Data Structure Name, Category, Structure and Definition. The user must be prompted via a popup box during the deletion process.

The prototype must use a two-dimensional array of type string to store each record. Refer to the program criteria to determine the exact size of the array, there is no provision for additional data! A double mouse click in the name text box will clear **all** four text boxes and focus the cursor into the name text box.

The wiki prototype will load and save data when the appropriate button is clicked, and all the wiki data is stored/retrieved using a binary file format. During the load and save process the end user must have the option to select an alternative data file. All end user interactions must have full error trapping and feedback messaging via a status strip at the bottom of the interface.

## Suggested Interface Design (example)

Table

Description automatically generated

## Question 1 Project Specifications

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 | | Data Structure Wiki | | | |
| Date | | 20/7/2023 | | | |
| Developer Name | | James Chellew | | | |
| Client Requirements | | | | | |
| Req. # | Description | | | Importance | Notes |
| 1 | Display list containing data structures for user to select. | | | 1(HIGH) | 12 rows 4 columns (from question 9) |
| 2 | 4 text boxes for the display of categories Data Structure Name, Category, Structure and Definition. | | | 1 | Definition should be multi-line text box |
| 3 | Selecting list item should populate the 4 text boxes | | | 1 |  |
| 4 | Searching data structure name should populate text boxes on a hit and retain focus on search text box. | | | 3 | Select item in Display list. Ensure proper casing of search. |
| 5 | On button press, add entries to display after populating the 4 text boxes. | | | 2 | Check if all fields are filled in or default the entry if there’s nothing. |
| 6 | On button press, apply edit and update any changes made in the 4 text box fields. | | | 2 | Prompt if field is empty or use default character like “~” |
| 7 | On button press, delete data in all 4 fields with confirmation. | | | 2 |  |
| 8 | On double click of item in display list, delete data in all 4 fields with confirmation. | | | 2 |  |
| 9 | Data should be stored in 2d arrays in the string data type | | | 1 | Update display box each time it is modified. |
| 10 | All data can be saved on button press. File will be saved in the format of a binary file. | | | 4 |  |
| 11 | Alternative data set to be loaded on button press. User should be able to select the data file they wish. | | | 4 |  |
| 12 | Program should include error trapping and user feedback text box to provide information back to the user. | | | 5(LOW) |  |
| 13 | Put tool tips on each element in program | | | 5 |  |
| User Interaction and Specifications | | | | | |
| GUI Component | | | Related Functions | | |
| List box | | | 1. Display Data of each row from wiki array (i.e. the three columns Name, category and structure of each entry) 2. Upon clicking/selecting list box item, display each column from the respective selected row in wiki array. 3. Update display when modifying or adding data. Clear selection after modifying. | | |
| Data structure name text box | | | 1. Display "Name" column of respective selected row from the list box. 2. Allow modification of this element that will update the wiki array element upon edit button push. | | |
| Category text box | | | 1. Display "Category" column of respective selected row from the list box. 2. Allow modification of this element that will update the wiki array element upon edit button push. | | |
| Structure text box | | | 1. Display "Structure" column of respective selected row from the list box. 2. Allow modification of this element that will update the wiki array element upon edit button push. | | |
| Definition text box (multi-line). | | | 1. Display "Definition" column of respective selected row from the list box. Information should be spread over multiple lines. 2. Allow modification of this element that will update the wiki array element upon edit button push. | | |
| Add button | | | 1. Will add an element to the wiki array if all (4) fields are filled with information. 2. Throw an error if not all fields filled. Displayed in error message text box. | | |
| Edit button | | | 1. Modify all fields of selected element. 2. Throw an error if modification leaves a field empty. | | |
| Delete button | | | 1. Delete selected element from 2D array. 2. Show message box to confirm element deletion. | | |
| Confirmation message box | | | 1. YES/NO message type message box prompting user if they are sure to delete the selected item. 2. Continue with deletion if YES, abort deletion if no. | | |
| Search text box | | | 1. Use input from this field to search the 2D array. 2. Upon hit, select the respective list box element (selection will trigger display of data) 3. Focus back on the search text box (have typing curser active) 4. Display unsuccessful search in user feedback text box. | | |
| Search button | | | 1. Use input from search text box to search 2D array upon button push. | | |
| Save button | | | 1. Display save file dialog on button push. | | |
| Savefiledialog | | | 1. By default, open to working directory. 2. Filter for \*.bin (binary file format) 3. When save is pressed in dialog box, save contents of 2D array into a binary file in the selected directory. 4. Cancel saving of file if the dialog is exited or cancel is pressed. | | |
| Open button | | | 1. Display open file dialog on button push. | | |
| Openfiledialog | | | 1. By default, open to working directory. 2. Filter for \*.bin files 3. When open is pressed, clear the 2D array, open selected file then re-populate the 2D array and list box from the file. | | |
| Labels | | | 1. Label each GUI element | | |
| User feedback and error message text box | | | 1. Display brief description of each action of user. 2. Display error message for blocked/invalid actions. | | |
| Tooltip | | | 1. Display description of each GUI element when hovering mouse over the respective element. | | |

# Question 2 Version Control Specifications

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: | | Git | VCS Name: | | Subversion (SVN) |
| URL: | | https://git-scm.com/ | URL: | | https://subversion.apache.org/ |
| Major Features | | | | | |
| List all the major features associated with each version control system (ref: https://www.softwaretestinghelp.com/version-control-software/) | | | | | |
| 1. | Uses distributed repository model. All developers store a copy of the repository locally and “sync” with centralised repository. | | 1. | Atomic, versioned commits. Changes are not made until the entire commit is successful and are recorded and stored per commit, not per file. | |
| 2. | Good source control with context switching. Experiment with ideas in an experimental branch, pull those ideas back into the main branch, then return to the experimental branch and test more new ideas. If new ideas don’t work, you can also dispose of the branch entirely. | | 2. | Each directory is versioned but not time stamped. Each version is treated as first-class objects, essentially like separate files.  First-class object: can be passed as an argument, returned from a function, or assigned to a variable. Reference: https://stackoverflow.com/questions/245192/what-are-first-class-objects | |
| 3. | Supports non-linear development. Can remove and add files seamlessly with support for rolling back to previous commits. | | 3. | Copying, renaming, moving and deleting is also versioned. | |
| 4. | Git operations are very quick in part due to a lot of the operations happening locally. It is several orders of magnitude faster than SVN in many common actions.  Reference: https://git-scm.com/about/small-and-fast | | 4. | Supports merge tracking. | |
| 5. | Multiple clones across devices as well as comprehensive history of changes made adds a great level of data redundancy. | | 5. | Supports file locking. Makes it so users require a special permission to modify a file. Allows everyone to view the file in read-only still. | |
| 6. | Offers good command line utility with git bash. | | 6. | Can work over network or local server. | |
| 7. | Integrated with common IDE’s | | 7. | Supports the creation of "slave repositories". A read-only version of the master repository that be synchronised with the master again later. | |
| 8. | Written in C | | 8. | Has greater windows support than alternatives like git. | |
| 9. | Open source and free | | 9. | Open source and free | |
| Recommendation: Which VCS would you choose and why? | | | | | |
| My recommendation would be git. Both GIT and SVN have great version tracking features, merging protection features and branching features but Git outclasses SVN in pull/push speed and other operations. Git also has great integration with the IDE to be used in this project, Visual studio 2022. Because of this, Git is the better choice for this project. | | | | | |

## Question 3 UML Class Details

Create a simple UML Class 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 Class Details template to answer question three.

|  |
| --- |
| UML Class Details |
| UML Diagram |
|  |
| Data Structure (use C# code) |
| static int row = 12;  static int column = 4;  string[,] wikiArray = new string[12,4] |

## Question 4 Development Methodologies

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: Waterfall |
| Description: Waterfall is a linear development methodology, each stage of development leads directly into the next stage, meaning the project does not support revising or changing the specifications or the scope of the project, all progress in the project moves towards software deployment. Each stage of the waterfall development style is well defined and strictly planned out such that there is no ambiguity in regards to the project, it needs to be so too as making a mistake or poor planning can prove detrimental as it is hard to "go back" and make fixes to the project. |
| Diagram: The 5 stages of the waterfall method process  Image from: https://business.adobe.com/blog/basics/waterfall |
| Advantages   1. Easy to manage. 2. Easy to follow. 3. Milestones are very blatant. 4. Each stage is well defined and usually well documented. |
| Disadvantages   1. Approach means that a working prototype will not be available until towards the end of cycle. 2. Cannot easy make changes once project moves forward. |
| 2nd Methodology Name: Agile |
| Description: The agile methodology splits the project up into smaller portions in different phases. Each phase is known as a "sprint". The agile methodology still includes client specification and planning, however, these specifications are generally loose and change is embraced with every iteration of the product and with new ideas. In the scrum agile framework, each small portion (or "issue" on Github) is displayed on a scrum board, and it is common that the portions of the project cycle back around the normal software development life cycle until it functions correctly. |
| Diagram:  Image From: https://www.researchgate.net/figure/Agile-Methodology-in-System-Development-source-Okeke2021-retrieved-from\_fig1\_354310848 |
| Advantages   1. Easy collaboration as each the project is split into parts. 2. A working prototype is produced earlier in the development life cycle. 3. Highly adaptable to changes in scope or client specifications. |
| Disadvantages   1. Due to the adaptable nature of agile, many things may not be documented. 2. Can be difficult to manage. 3. Development may alter the overall function through iterations |
| 3rd Methodology Name: Iterative |
| Description: The iterative methodology focuses on steady improvement and enhancement of the software through creating many prototypes and iterations of it. It will usually start with a foundation, a prototype, with minimal features and iteratively improves upon the design and feature set through testing and evaluating previous implementations with each version until final implementation. |
| Diagram: undefined |
| Advantages   1. Working prototypes are produced early in the development life cycle 2. Great flexibility. The supports changing the scope of the project as functionality can be implemented and removed from each iteration. 3. Bugs can be easier to find and fix as testing is done on each iteration of the program. 4. New issues or requirements can be addressed quickly and easily. |
| Disadvantages   1. There is not clear "end point" for the software making it hard to judge how long development will take. 2. Can be hard to manage an ever-changing product. 3. Edge case issues my arise due to scope changes not accounted for at the beginning of development (System architecture or conflicting design etc.) 4. Not very suitable for smaller projects. |
| 4th Methodology Name: Spiral |
| Description: The spiral methodology is best described as a combination of the prototyping style of the iterative methodology with the systematic aspects of the waterfall methodology. The "spiral" consists of 4 main parts, determining objectives, identifying and solving risks, development the next iteration, reflecting and reviewing the iteration for the next loop. |
| Diagram: Lightbox |
| Advantages   1. Can effectively handle risks or issues that may present themselves throughout development. 2. Great flexibility due to its iterative aspect. 3. Working prototype is produced earlier in development life cycle. 4. Incremental approach allows the functionality and features to improve and evolve with each iteration. |
| Disadvantages   1. Not suited for smaller projects. 2. Very dependant on risk analysis to progress. 3. Like the iterative methodology, there is not clear "end point" to the project making time management difficult. 4. Complexity of the project can scale very quickly. |
| References |
| Waterfall diagram image:  <https://business.adobe.com/blog/basics/waterfall>  Agile information:  <https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/>  Agile Diagram image:  <https://www.researchgate.net/figure/Agile-Methodology-in-System-Development-source-Okeke2021-retrieved-from_fig1_354310848>  Iterative Information:  <https://www.tutorialspoint.com/sdlc/sdlc_iterative_model.htm>  Iterative diagram image: <https://en.wikipedia.org/wiki/Iterative_and_incremental_development#/media/File:Iterative_Process_Diagram.svg>  Spiral information and diagram image:  <https://www.geeksforgeeks.org/software-engineering-spiral-model/> |

## Question 5 Methodology Analysis and Selection

Refer back to the previous question 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 |
| Must be suitable for small projects |
| Due to the nature of my specific skill set (being a programming novice), the project would benefit if the methodology was adaptable or flexible for new ideas. |
| Would benefit if project was separated into small parts to focus on |
| Methodology should be open to new ideas or more creative solutions down the line |
| Testing should be performed throughout the cycle of the project to ensure all functions work and interact correctly |
| Methodology should support changing of specification priorities |
| Methodology Selection |
| Methodology Name Agile |
| Justification (reasons for selection) |
| Agile can be suitable for almost any size project |
| Changes and adaptations can be made after each sprint, allowing for new ideas or adapting the program |
| Having each function separated into different todo items allows for one, focus to be solely on the one item, and two, allows a priority shift should the project require it. |
| Testing is done at the completion of each todo item, allowing bugs to be discovered and fixed as they happen. |
| Personal preference – working on each "todo" item separately helps keep me more on task and efficient. |

## Question 6 Manage Version Control System

CITE uses GitHub as the primary Version Control System (VCS), create your own GitHub account (you can use your existing account). Ensure your development computer has Visual Studio installed with all the appropriate features to create and develop a C# .Net application. Open Visual Studio and create a new project; ensure the GitHub settings have been updated to reflect your GitHub account. Create a new repository for the project and check you have both a local and remote (Cloud) instance of the repository.

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 | | Git (GitHub) | | | | |
| Version Details | | 2023 | Date | 02/08/2023 | | |
| Supported Platforms | | Windows, Mac OS, Linux | | | | |
| Repository/Project Information | | | | | Yes | No |
| Has the cloud VCS account been created and named correctly? | | | | |  |  |
| Has the basic project solution been created? | | | | |  |  |
| Has the cloud VCS repository been created | | | | |  |  |
| Does the cloud VCS repository have a readme file? | | | | |  |  |
| Has the local source repository been created? | | | | |  |  |
| Are the two VCS resources linked? | | | | |  |  |
| Account Details Checklist | | | | | | |
| Repository Name: | DataStructureWikiForm | | | | | |
| URL | https://github.com/JamesChellew/DataStructureWikiForm | | | | | |
| Local Source Control Screen Shot |  | | | | | |
| Cloud Screen Shot |  | | | | | |

## Question 7 Project Plan

Using your recommended development methodology from the previous question, 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. Using the GitHub repository from the Manage VCS question create a Project within the repository and add your project plan.

|  |  |  |  |
| --- | --- | --- | --- |
| Project Plan | | | |
| Project Name | Data Structure Wiki | | |
| Date | 02/08/2023 | | |
| Developer Name | James Chellew | | |
| Development Tasks | | | |
| Task Name | Task Type | Task Description | Input/Output Parameters |
| Analysis of client requirements | analysis | Analyse client requirements and develop the User interaction specifications document (specifications of everything the user interacts with) and the client requirements/project specifications document (specifications of how the program should run) | User interaction docs and project specs. |
| Compare version control systems | research | Research and curate the features of two different version control systems, one being Git/Github | Vcs comparison doc, online resources |
| Comparing development life cycle methodologies | research | Research and compare the advantages and disadvantages of 4 software development life cycle methodologies. Select the one most suited for this project | Development life cycle comparison doc and choice of methodology, online resources. |
| Set up version control | Version control | Set up GitHub repository with appropriate name. | Cloud repository |
| Code | development | Develop program according to user interactions document and project specifications document. | Project specs, user interaction document, github, c# api docs |
| ui | development | Develop UI based on user interactions document | User interactions document |
| Test and return to coding | testing | Test code, identify bugs and return to coding stage to fix. | dataStructureWikiForm program |
| Testing docs | documentation | Produce testing documentation including screenshots of the code | Debugging document |
| demonstration | presenting | Present and demonstrate the program to the lecturer | Online presentation |

## Question 8 Sign-off and Approval

You will need to arrange for the previous completed questions to be reviewed by the Lecturer/Assessor for sign off, approval and feedback before you start the development.

Question 1 – Project Specifications

Question 2 – Version Control Specifications

Question 3 – UML Class Details

Question 4 – Development Methodologies

Question 5 – Methodology Analysis and Selection

Question 6 – Manage Version Control System

Question 7 – Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
|  |  |  |  |  |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
|  | | | | |

## Question 9 Prototype Development

Create the wiki prototype to demonstrate how a collection of information can be stored using a Windows Application (WinForms). 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 the previous Question to manage your code during the development; ensure you record these commits/branches as a series of screen shots to be included in Version Update Question (ie start, working, final). Your code must adhere to the CITEMS software development standards. (refer http://www.citems.com.au/)

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

The following programming criteria and features are required, check the original project specifications for additional information.

### Programming Criteria

1. Create a global 2D string array, use static variables for the dimensions (row = 12, column = 4),
2. Create an ADD button that will store the information from the 4 text boxes into the 2D array,
3. Create an EDIT button that will allow the user to modify any information from the 4 text boxes into the 2D array,
4. Create a DELETE button that removes all the information from a single entry of the array; the user must be prompted before the final deletion occurs,
5. Create a CLEAR method to clear the four text boxes so a new definition can be added,
6. 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 the array element to be swapped (do not use any built-in array methods),
7. 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),
8. Create a display method that will show the following information in a ListView: Name and Category,
9. Create a method so the user can select a definition (Name) from the ListView and all the information is displayed in the appropriate Textboxes,
10. 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,** ensure theuser has the option to select an alternative file. Use a file stream and BinaryWriter to create the file.
11. Create a LOAD button that will read the information from a binary file called ***definitions.dat*** into the 2D array, ensure theuser has the option to select an alternative file. Use a file stream and BinaryReader to complete this task.
12. All code is required to be adequately commented, and each interaction must have suitable error trapping and/or feedback. All methods must utilise the appropriate Dialog Boxes, Message Boxes, etc to ensure fully user functionality. Map the programming criteria (9.1 - 9.11) 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 10 Version Control Update

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: | DataStructureWikiForm |
| URL | https://github.com/JamesChellew/DataStructureWikiForm |
| Desktop Screen Shots |  |
| Cloud Screen Shots |  |

## Question 11 Data Structure Matrix

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 | An array is a collection of elements – elements being values or variables – of the same data type. In memory, arrays are stored in "contiguous" memory locations, each of the elements are stored in memory that are in proximity of each other with the first element occupying the lowest memory location. Arrays have an index, the location of an element in the array, and an address, the location of the element in memory. When coding, you would simply use the index of an element to access it. In memory however, the element's position is calculated using the base address (starting position in memory) and the size of the data elements. |
| Two Dimension Array | Linear | A two-dimensional array, or 2D array, is an array that is structured like a matrix with rows and columns in a grid structure. Each element in an array can be referred to as a cell, which can be accessed using its respective row and column index. |
| List | List | Linear | A list is an ordered collection of elements. A list allows repetition of data and can also store a heterogenous mix of different data types within the same list. Each element of data is treated as a unique entry. A list is also not fixed in size like an array, data can be added and removed freely without needing to specify the size. |
| Linked list | Linear | A linked list is a linear collection of data elements like that of a normal list. The difference comes with how the linked list is treated in memory. Each "node", an element/pointer pairing, in the list contains the data element and a pointer pointing to the next node in the computer's memory. By extension, this means that the data elements are not stored together in memory. |
| Self-Balance Tree | Tree | Non-Linear | A self-balance tree is the same as something like a binary search tree. The difference being that a self-balance tree automatically sorts itself such that it keeps its height to the minimum possible. |
| Heap | Non-Linear | A heap is a rooted tree diagram that satisfies the heap property. The heap property is where each child nodes are smaller than the parent or key nodes, or where the child nodes are greater than the parent or key nodes. This is called a "max heap" and a "min heap" respectively. |
| Binary Search Tree | Non-Linear | A binary search tree is a rooted tree data structure with each parent node being greater than all values on the left sub-tree, and smaller than all values on the right sub-tree. The key or search term is the root of all the nodes. The efficiency of this tree for is proportional to the height of the tree (i.e. the number of parent nodes) |
| Graph | Graphs | Non-Linear | A graph is represented by a set of nodes or vertices that are linked together in sets of pairs. The nodes represent any form of data with the links, known as edges, linking the two nodes to each other. Adjacency refers the link between two nodes; nodes are adjacent if there is an edge connecting them. Path refers to the sequence of edges that connects two nodes. Graphs can be directed or undirected. Edges in an undirected graph link nodes both ways. Edges in a directed graph may be one or two ways. |
| Set | Abstract | Non-Linear | A set is an unordered collection of unique values. That is, elements in a set do not have a particular order and each value within the set does not repeat. |
| Queue | Linear | A queue an object that contains a collection of elements or entities that follows a first-in-first-out rule. That is, the first element that goes in the queue, or enqueued, is that first element that goes out of the queue, or dequeued. In other words, new entities start at the back of the queue and entities are dropped from the front working through the queue. |
| Stack | Linear | A stack an object that contains a collection of elements or entities that follows a first-in-last-out rule. That is, the first element that goes into a stack, referred to as a push, is the last one to go out of the stack, referred to as a pop. In other words, a stack is like picking entities out of a pringles tin, the last one you put on (pushed) the stack is that first one that will get pulled out (popped). |
| Hash Table | Hash | Non-Linear | A hash table is a collection of key-value pairs. Keys are unique integers that indexes the values. Values are the data elements that relate to the keys. |
| References | | | |
| Array:  <https://www.javatpoint.com/data-structure-array>  <https://www.simplilearn.com/tutorials/data-structure-tutorial/arrays-in-data-structure>  <https://en.wikipedia.org/wiki/Array_(data_structure)>  Two Dimensional Array:  <https://www.freecodecamp.org/news/javascript-2d-arrays/#:~:text=A%20two%2Ddimensional%20array%2C%20also,row%20and%20column%20indices%2Findexes>.  List:  <https://www.javatpoint.com/list-data-structure>  Linked List: <https://en.wikipedia.org/wiki/Linked_list>  Self-balance tree:  <https://en.wikipedia.org/wiki/Self-balancing_binary_search_tree>  Heap:  <https://www.programiz.com/dsa/heap-data-structure>  Binary Search tree:  <https://en.wikipedia.org/wiki/Binary_search_tree>  Graph:  <https://www.programiz.com/dsa/graph>  <https://en.wikipedia.org/wiki/Graph_(abstract_data_type)>  Set:  <https://www.programiz.com/python-programming/set>  <https://en.wikipedia.org/wiki/Set_(abstract_data_type)>  Queue:  <https://www.programiz.com/dsa/queue>  <https://en.wikipedia.org/wiki/Queue_(abstract_data_type)>  Stack:  <https://www.simplilearn.com/tutorials/data-structure-tutorial/stacks-in-data-structures#:~:text=The%20stack%20data%20structure%20is,of%20money%2C%20and%20many%20more>.  Hash Table:  <https://www.programiz.com/dsa/hash-table> | | | |

## Question 12 UML Activity Diagram

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 pahways are tested. Update the C# Code on the right section to identify the breakpoints. The example in the appendix is provided for clarification. Complete the following UML Activity Diagram template below.

|  |  |
| --- | --- |
| UML Activity Diagram | |
| Diagram | C# Code |
|  | private void ButtonSearch\_Click(object sender, EventArgs e)  {  int startIndex = 0;  int endIndex = maxRows;  int midIndex = 0;  bool flag = false;  string input = TextBoxSearch.Text;  if (!String.IsNullOrWhiteSpace(input))  {  while (!flag && startIndex < endIndex)  {  midIndex = (endIndex + startIndex) / 2;  if (input == wikiArray[midIndex, 0])  {  flag = true;  break;  }  else if (String.IsNullOrWhiteSpace(wikiArray[midIndex, 0]))  {  endIndex = midIndex - 1;  }  else if (input.CompareTo(wikiArray[midIndex, 1]) > 0  {  startIndex = midIndex + 1;  }  else  {  endIndex = midIndex - 1;  }  }  if (flag)  {  ListViewWiki.SelectedItems.Clear();  ListViewWiki.Items[midIndex].Selected = true;  TextBoxFeedback.Text = "Search result: " + input + ", was found";  TextBoxSearch.Clear();  TextBoxSearch.Focus();  }  else  {  TextBoxFeedback.Text = "No matching results";  ListViewWiki.SelectedItems.Clear();  TextBoxSearch.Focus();  }    }  } |

## Question 13 Debug Test Report

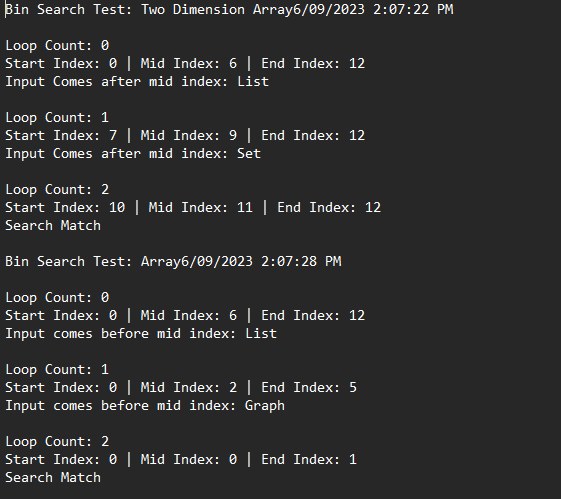
Using the breakpoints shown in the previous Question as a starting point, utilise the debug features to debug, trace and test your Binary Search code. Ensure your code is error free and functions correctly (refer Programming Criteria) record and correct any errors. Your Debug Test Report must include 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 previous Question 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 | | Data Structure Wiki Form | | | | | | | | |
| **Method** | | Trace | | | | | | | | |
| **Description** | | Trace testing binary search method. Outputs trace information to txt file. | | | | | | | | |
| **Level of Testing** | |  | | | | | | | | |
| **Developer** | | James Chellew | | **Tester** | James Chellew | | **Date** | | 05-09-2023 | |
| **Test Case No** | **Test Case Name** | | **Test steps** | | | **Test Data** | | **Expected result** | | **Pass / Fail** |
| 1 | Last index in array | | Type "Two Dimension Array" then press search button | | | Two Dimension Array selected | | Two Dimension Array selected | | pass |
| 2 | First index in array | | Type "Array" then press search | | | Array selected | | Array selected | | pass |
| 3 | Mid index in array | | Type "Linked List" then press search | | | Linked List | | Linked list selected. | | pass |
| 4 | No match | | Type "This should not match" then press search. | | | Not item Selected and message displayed | | No item selected and message displayed for no match | | pass |
| 5 | Case sensitivity | | Type "array" and the press search. | | | No item selected and message displayed | | Array selected | | fail |
| 6 | Whitespace | | Type " " and then press search. | | | No item selected and message displayed | | Error displayed to user | | fail |
| 7 | Whitespace at end | | Type "Array " then press search. | | | No item selected and message displayed | | Array selected | | fail |

*Test Case 1:*

*A screenshot of a computer

Description automatically generated* 

*Test Case 2:*

*A screenshot of a computer

Description automatically generated* A screenshot of a computer program

Description automatically generated

*Test Case 3:*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated*

*Test Case 4:*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer program

Description automatically generated*

*Test Case 5:*

*A screenshot of a computer

Description automatically generated*

*A screenshot of a computer

Description automatically generated*

*Test Case 6:*

*A screenshot of a computer

Description automatically generated*

**

*Test Case 7:*

*A screenshot of a computer

Description automatically generated*

*A black screen with white text

Description automatically generated*

## Question 14 Post Development Analysis

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 Software Development Life Cycle 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?  I used the Agile methodology. |
| How effective was your project plan in developing the final prototype?  My project plan was very effective. It gave me enough of a guide for me to one, stay on task and have a clear vision of the next steps I needed to take in order to complete the project, and two, was loose enough so that I could work flexibly on the tasks I had assigned myself. It really allowed me to experiment and then later optimise my code, giving me tangible prototypes to play with and visualise the code I had written. |
| What alternative data structures could be used?  A list could have been used in the place of an array. This would allow the collection of data to be more flexible, most notably in size. Using list would make it easier to enter add and remove data from the wiki. Each data structure could also have its information stored in an class. |
| What constructive advice could you provide for the development of a similar project?  Separate and organise your code. Write separate methods for each specific operation to increase readability and coding efficiency. Writing methods separately for each operation will allow you to reuse the code you have already written, reducing repetition which saves you time. Also, ensure you make many git commits so that if you break something and are not sure how to fix it, you can just roll back to a previous commit and try again. |

## Question 15 Demonstration and Submission

Demonstrate your working program to your lecturer using the realistic data from the previous Question. 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.

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

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

End of Assessment Task One

## Appendix

Diagram

Description automatically generated