|  |  |  |  |
| --- | --- | --- | --- |
| **Qualification details** | | | |
| **Training Package Code and Title** | ICT - Information and Communications Technology (Version 8.0) | | |
| **Qualification National Code and Title** | ICT50220 Diploma of information Technology (Release 2) | **State code** | BGJ4 |
| **Assessment Title** *(as per DAP)* | Assessment Project One (Individual Project) | | |
| **Unit National Code & Title** | ICTPRG535 Build advanced user interfaces | | |
| ICTPRG547 Apply advanced programming skills in another language | | |
| ICTICT517 Match ICT needs with the strategic direction of the organisation | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Date Due** | Week Five | | **Date Received** | |  | |
| **Student Name** |  | | | | | |
| **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 Project |
| Duration of the assessment | 5 class sessions (Weeks 1 - 5) |
| 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:  ICTPRG5335 Build advanced user interfaces   * Plan and design a UI solution according to organisational requirements, * Apply interactions designs and implement validation requirements against the design plan, * Create and display the UI with graphics according to UI requirements.   ICTPRG547 Apply advanced programming skills in another language   * Code advanced data structures using hashing, sorting and searching algorithms, * Apply third party libraries and communication technologies for data exchange, * Test and evaluate the code to resolve logical and syntactical designs flaws, * Create and document the application according to technical specifications.   ICTICT517 Match ICT needs with the strategic direction of the organisation   * Interpret, analysis and report the strategic organisational plan * Propose and document changes for the implementation of a ICT system * Provide action plan and schedule   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. |

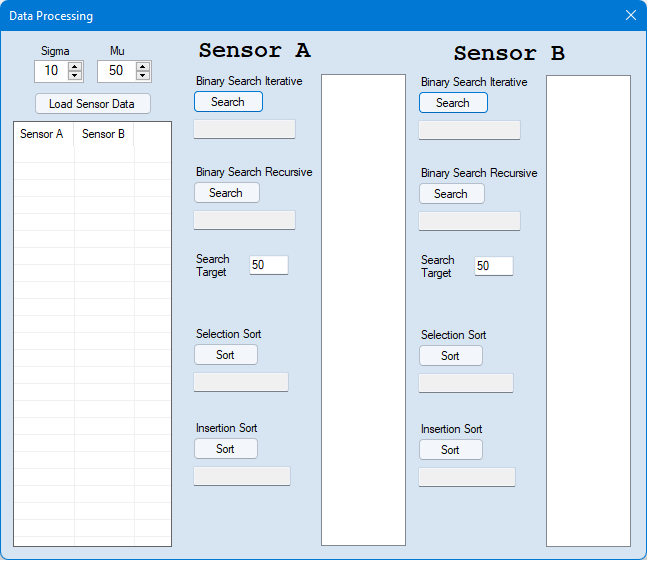
|  |  |
| --- | --- |
| Allowable Materials | Blackboard Topics, SDLC, Weekly readings (PDF), Example programs and Independent Outside of Class Activities |
| Required Resources | Web links and example code can be downloaded from the Blackboard portal.  PC with Notepad++, Visual Studio, GitHub, MSOffice.  Internet Access to 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 web coding 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 – Design Specification  Question 2 – Version Control  Question 3 – Design Approval  Question 4 – Application Development  Question 5 – Debug and Testing  Question 6 – Evaluation  Question 7 – Assessment, Feedback and Signoff |

# Scenario

You are employed as the Senior Programmer with CITE Managed Services and have been assigned the Satellite Data Processing Project for an organisation called Malin Space Science Systems (MSSS). This project will require the planning, development, testing and reporting of an application that will sort and search complex data sets recorded during satellite operations. The senior scientists at Malin would like two sorting and two searching algorithms so they can compare the processing of raw data as it is collected. This application will utilise two advanced sorting algorithms and two binary searching techniques. The details and criteria are provided in the following paragraphs.

## Background Information

Marlin Space Science Systems operate a satellite with a Sun-synchronous orbit, which means they can record data from the same spot at the same local time each day. The advantage for scientists means they can compare data from a fixed location over a long period of time to detect changes. The data is collected as raw numbers from two sensors on the satellite as it passes over the receiving station. This stream of numbers is then transmitted and saved onto a server for further examination. The processing occurs when each data stream is read into the application using a mathematical formula embedded in a Dynamic Link Library called Galileo. Currently the scientists use a 32bit command line application which is executed from a console, this old software is not able to run successfully on new hardware. There has been a series of recording that are outside the expected range of normal readings and the scientists at Malin require a replacement application which could provide accurate information. Therefore, the organisation has requested a Windows based application that is compatible with contemporary hardware. The analysis team have consulted with Marlin and provided an interface layout for the display of satellite information with the required features.



Ref: <https://en.wikipedia.org/wiki/Sun-synchronous_orbit>

Ref: <https://www.esa.int/ESA_Multimedia/Images/2020/03/Polar_and_Sun-synchronous_orbit>

## Application Requirements

The Marlin scientists require a Windows Application to process the data using the Galileo DLL and two LinkedList’s; one for each satellite sensor. The data from each sensor must be read into a simple LinkedList of type “double” and then sorted using a Selection and Insertion sort algorithm. Next, the scientist will enter a value into a search textbox and select a Recursive or Iterative binary search algorithm. Finally, for each of the four algorithms the processing time must be measured and displayed.

The Marlin scientists would like buttons for each action and several display options to examine the raw data and processed information. The application must have a single form that shows all the information from the two sensors. The following features are mandatory:

* A ListBox to display the data from each Sensor.
* A ListView with two columns to display the data from both sensors (Sensor A and Sensor B).
* A numeric input to control the standard deviation (Sigma) and mean (Mu) of the data.
* A button to load data into the two LinkedLists and then display these LinkedLists in a ListView.
* A series of buttons to process the sort and search functions.
* Textbox display for the processing times each sort and search.

You should consult with the CITE representative (your Lecturer) if you are unsure about any of the problems or questions in this assessment. Your primary research should focus on the resources on the Blackboard LMS and CITE web site, additional information can be collected from the Internet, ensure all sources are referenced in your submission. Ensure your development follows an Agile methodology that is recorded and maintained using your GitHub account. You must demonstrate your working applications before uploading to Blackboard, your Lecturer (Assessor) will sign off to ensure all the criteria are satisfied.

## Question 1 Design Specification

Provide a suitable description/explanation for each client requirement, and then complete the UI design specification which outlines all the user interactions that occur when buttons and related components are clicked or selected. Complete the following Design Specification template to answer this question.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Specification | | | | |
| Developer Name | Kyle Watson | | Date | 18/07/2022 |
| Client Technical Requirements | | | | |
| Requirements | | Description | | |
| 1. 2x LinkedList<double> | | Two LinkedList data structures of type ‘double’ declared as global within the public partial class. | | |
| 1. 2x ListBoxes | | ListBoxes for displaying the data from both sensors | | |
| 1. ListView | | ListView has two columns for displaying data from both sensors | | |
| 1. Numeric input control (Sigma) | | Controls the standard deviation of the data | | |
| 1. Mean control (Mu) | | Controls the mean of the data | | |
| 1. Load Button | | Loads data into the two LinkedLists and displays them in a ListView | | |
| 1. 4x Sort button | | Processes for the sort functions for sensors A and B | | |
| 1. 4x Search button | | Processes for the search functions for sensors A and B | | |
| 1. 8x Read only boxes | | 4x read only boxes that display the number of ticks required to complete each search function.  4x read only boxes that display the number of milliseconds required to complete each sort function. | | |

|  |  |
| --- | --- |
| UI Design Specification  What are the User Interactions when a form component is clicked or select? | |
| UI Action | Description |
| 4x button click search methods | Four button click methods that will search the LinkedList for a value entered into a textbox on the form   1. Method for Sensor A - Binary Search Iterative 2. Method for Sensor A - Binary Search Recursive 3. Method for Sensor B - Binary Search Iterative 4. Method for Sensor B - Binary Search Recursive |
| 4x button click sort methods | 1. Method for Sensor A - Selection Sort 2. Method for Sensor A - Insertion Sort 3. Method for Sensor B - Selection Sort 4. Method for Sensor B - Insertion Sort |
| Load & Display button | Button that will call the LoadData and ShowAllSensorData methods |
| Numeric up down controls | For incrementally increasing and decreasing the value of the mean and standard deviations |
| TextBoxInput for mean and standard deviation | Input boxes for declaring the mean and standard deviation |

## Question 2 Version Control

Malin Space Science Systems would like you to use GitHub as the primary source control, setup an appropriate structure in your GitHub account to manage the Astronomical Processing Project development. Add a kanban project to your repository which reflects the basic Agile development process you intend to pursue. Complete the following GitHub Version Control template to answer this question.

|  |  |  |  |
| --- | --- | --- | --- |
| GitHub Version Control | | | |
| Repository Name: | Malin-Space-Science-System | | |
| URL | https://github.com/Kwatson-1/Malin-Space-Science-System | Date | 18/07/2022 |
| Screen Shot(s) |  | | |

## Question 3 Design Approval

Once you have complete questions 1 & 2 arrange for your document to be reviewed by the Lecturer/Assessor for approval, sign off and feedback before completing the development and testing.

* Question 1 Design Specification
* Question 2 Version Control

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Design Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
| Stewart Godwin | Lecturer |  | 18/07/2022 | Yes |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
| Include clear method each time load method is called. | | | | |

## Question 4 Application Development

When you have received approval from the client (or Lecturer) your next task is to create the Windows Application which satisfies the following criteria and uses the supplied Galileo dll. The exact requirements are critical as the Malin scientists are using this application to determine anomalies in data collected over the Mount Etna volcano in Italy. As you develop the application ensure you debug and test each algorithm to ensure compliance with the client’s criteria.

**Important:** You can only declare two global variables which represent the LinkedLists for Sensor A and Sensor B. You are not permitted to use any other global variables; all methods should utilise input parameters and returns types in order to process information.

**Global Methods**

1. Create two data structures using the LinkedList<T> class from the C# Systems.Collections.Generic namespace. The data must be of type “double”; you are not permitted to use any additional classes, nodes, pointers or data structures (array, list, etc) in the implementation of this application. The two LinkedLists of type double are to be declared as global within the “public partial class”.
2. Copy the Galileo.DLL file into the root directory of your solution folder and add the appropriate reference in the solution explorer. Create a method called “LoadData” which will populate both LinkedLists. Declare an instance of the Galileo library in the method and create the appropriate loop construct to populate the two LinkedList; the data from Sensor A will populate the first LinkedList, while the data from Sensor B will populate the second LinkedList. The LinkedList size will be hardcoded inside the method and must be equal to 400. The input parameters are empty, and the return type is void.
3. Create a custom method called “ShowAllSensorData” which will display both LinkedLists in a ListView. Add column titles “Sensor A” and “Sensor B” to the ListView. The input parameters are empty, and the return type is void.
4. Create a button and associated click method that will call the LoadData and ShowAllSensorData methods. The input parameters are empty, and the return type is void.

**Utility Methods**

1. Create a method called “NumberOfNodes” that will return an integer which is the number of nodes(elements) in a LinkedList. The method signature will have an input parameter of type LinkedList, and the calling code argument is the linkedlist name.
2. Create a method called “DisplayListboxData” that will display the content of a LinkedList inside the appropriate ListBox. The method signature will have two input parameters; a LinkedList, and the ListBox name. The calling code argument is the linkedlist name and the listbox name.

**Sort and Search Methods**

1. Create a method called “SelectionSort” which has a single input parameter of type LinkedList, while the calling code argument is the linkedlist name. The method code must follow the pseudo code supplied below in the Appendix. The return type is Boolean.
2. Create a method called “InsertionSort” which has a single parameter of type LinkedList, while the calling code argument is the linkedlist name. The method code must follow the pseudo code supplied below in the Appendix. The return type is Boolean.
3. Create a method called “BinarySearchIterative” which has the following four parameters: LinkedList, SearchValue, Minimum and Maximum. This method will return an integer of the linkedlist element from a successful search or the nearest neighbour value. The calling code argument is the linkedlist name, search value, minimum list size and the number of nodes in the list. The method code must follow the pseudo code supplied below in the Appendix.
4. Create a method called “BinarySearchRecursive” which has the following four parameters: LinkedList, SearchValue, Minimum and Maximum. This method will return an integer of the linkedlist element from a successful search or the nearest neighbour value. The calling code argument is the linkedlist name, search value, minimum list size and the number of nodes in the list. The method code must follow the pseudo code supplied below in the Appendix.

**UI Button Methods**

1. Create four button click methods that will search the LinkedList for a value entered into a textbox on the form. The four methods are:
   1. Method for Sensor A and Binary Search Iterative
   2. Method for Sensor A and Binary Search Recursive
   3. Method for Sensor B and Binary Search Iterative
   4. Method for Sensor B and Binary Search Recursive

The search code must check to ensure the data is sorted, then start a stopwatch before calling the search method. Once the search is complete the stopwatch will stop and the number of ticks will be displayed in a read only textbox. Finally, the code will call the “DisplayListboxData” method and highlight the appropriate number (or the next closest number).

1. Create four button click methods that will sort the LinkedList using the Selection and Insertion methods. The four methods are:
   1. Method for Sensor A and Selection Sort
   2. Method for Sensor A and Insertion Sort
   3. Method for Sensor B and Selection Sort
   4. Method for Sensor B and Insertion Sort

The button method must start a stopwatch before calling the sort method. Once the sort is complete the stopwatch will stop and the number of milliseconds will be displayed in a read only textbox. Finally, the code will call the “ShowAllSensorData” method and “DisplayListboxData” for the appropriate sensor.

1. Add two NumericUpDown controls for Sigma and Mu. The value for Sigma must be limited with a minimum of 10 and a maximum of 20. Set the default value to 10. The value for Mu must be limited with a minimum of 35 and a maximum of 75. Set the default value to 50.
2. Add two textboxes for the search value; one for each sensor, ensure only numeric values can be entered.
3. 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/).

## Question 5 Debug and Testing

Ensure your code is error free and functions correctly, then design and test the applications using several different sets of data. During these tests check the returned times are record correctly and formatted into the appropriate scientific units. Your Test Report must include appropriate evidence that your application functions as expected (references to screen captures).

Finally, test the application to ensure all the user interactions are working as expected. Complete the following Test Report template to answer this question.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Report | | | | | |
| Developer Name |  | | **Date** |  | |
| Sensor Data Test | | | | | |
| Input Data | Description | Actual Result | | | Pass / Fail |
|  |  |  | | |  |
|  |  |  | | |  |
|  |  |  | | |  |
|  |  |  | | |  |
|  |  |  | | |  |
| User Experience Tests | | | | | |
| UI Component | Description | | | | Pass / Fail |
|  |  | | | |  |
|  |  | | | |  |
|  |  | | | |  |
|  |  | | | |  |
|  |  | | | |  |

## Question 6 Evaluation

Once testing has been completed you can answer the following questions regarding the implementation and usage of your application.

|  |  |  |  |
| --- | --- | --- | --- |
| Evaluation Report | | | |
| Developer Name |  | **Date** |  |
| Reflective Questions | | | |
| Your Windows application will replace the current 32Bit command line program, What are the impacts of this change? | | | |
| What are Malin’s objectives for requesting the new software application? | | | |
| How will the Forms based application meet the objectives of the Malin organisation? | | | |
| What are the perceived difficulties of introducing the new Forms based application? | | | |
| What training could be required to use the new Forms based application? | | | |
| What time schedule would you recommend for the implementation of training. | | | |
| How would you seek feedback on the success or failure of your Forms based application? | | | |

## Question 7 Assessment, Feedback and Signoff

Check all the above documentation has been completed and is ready for inspection. Contact your Lecturer (Assessor) and arrange to demonstrate your working applications, use the following Marking Guide and Observation Checklist to ensure you have completed all the assessment criteria.

### Assessor Marking Guide

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Marking Guide and Observation Checklist | | Satisfactory | | Feedback |
| **Questions** | | YES NO | |  |
| Q1 | Design Specification: All fields of the Design Specification are filled in. |  |  |  |
|  | Client Technical Requirements contains information that is correct. |  |  |  |
|  | UI Specifications show a detailed information when a form component is clicked or select? |  |  |  |
| Q2 | Version Control: All fields of the template are filled in. There are screen shots of GitHub showing the Project and Repository. |  |  |  |
|  | Lecturer has observed the GitHub account and it reflects an Agile project methodology. |  |  |  |
|  | Lecture has observed the GitHub account which reflects a repository with appropriate files. |  |  |  |
| Q3 | Design Approval has been signed off by Lecturer |  |  |  |
|  | Suitable feedback has been provided |  |  |  |
|  | Anomalies have been corrected |  |  |  |
| Q4 | There are only two global variables (two LinkedLists) |  |  |  |
|  | The DLL has been referenced in the solution. The data from each sensor has been added to the relative LinkedList. All criteria in 4.2 are correct. |  |  |  |
|  | The sensor data can be displayed in a ListView and called from a Button click method |  |  |  |
|  | The two utility methods have the correct input parameters. |  |  |  |
|  | The sort and search methods have the correct algorithm. Each input parameter is correct and all criteria in 4.7 - 4.10 are correct. |  |  |  |
|  | The four UI buttons for the searches have the correct “calls” and return the correct time units. |  |  |  |
|  | The four UI buttons for the sorts have the correct “calls” and return the correct time units. |  |  |  |
|  | The form controls meet the 4.13 – 4.15 criteria, All code is correctly commented |  |  |  |
| Q5 | Testing: All the fields in the Testing Report have been filled in. |  |  |  |
|  | The sensor test data has tested each search and sort for both sensors, (8 tests) |  |  |  |
|  | Each test has been run more than once. |  |  |  |
| Q6 | Evaluation: The evaluation questions have been answered. |  |  |  |
| Q7 | Feedback: The client (Lecturer) has provided suitable feedback for the assessment task. |  |  |  |
|  | Feedback: The client (Lecturer) has observed the Github project and completed development plan. |  |  |  |
| **General Feedback:** | | | | |
|  | **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 One

## Appendix

### Selection Sort

integer min => 0

integer max => numberOfNodes(list)

for ( i = 0 to max )

min => i

for ( j = i + 1 to max )

if (list element(j) < list element(min))

min => j

END for

**// Supplied C# code**

LinkedListNode<double> currentMin = list.Find(list.ElementAt(min))

LinkedListNode<double> currentI = list.Find(list.ElementAt(i))

**// End of supplied C# code**

var temp = currentMin.Value

currentMin.Value = currentI.Value

currentI.Value = temp

END for

### Insertion Sort

integer max = numberOfNodes(list)

for ( i = 0 to max – 1 )

for ( j = i + 1 to j > 0, j-- )

if (list element(j - 1) > list element(j))

**// Supplied C# code**

LinkedListNode<double> current = list.Find(list.ElementAt(j))

**// End of supplied C# code**

**// Add Swap code here by swapping**

**// current previous value with current value.**

END if

END for

END for

### Binary Search Iterative

while (minimum <= maximum - 1)

integer middle = minimum + maximum / 2

if (search value = list element(middle))

return ++middle

else if (search value < list element(middle))

maximum => middle - 1

else

minimum => middle + 1

END while

return minimum

### Binary Search Recursive

if (minimum <= maximum - 1)

integer middle = minimum + maximum / 2

if (search value = list element(middle))

return middle

else if (search value < list element(middle))

return binarySearchRecursive(list, search value, minimum, middle - 1)

else

return binarySearchRecursive(list, search value, middle + 1, maximum)

END if

return minimum

### Galileo DLL

**<?xml version="1.0"?>**

**<doc>**

**<assembly>**

**<name>**Galileo**</name>**

**</assembly>**

**<members>**

**<member** **name="T:Galileo.ReadData">**

**<summary>**

A data recording library that returns the readings from two sensors.

**</summary>**

**</member>**

**<member** **name**=**"M:Galileo.ReadData.SensorA(System.Double,System.Double)">**

**<summary>**

Sensor A records data that represents a Gaussian distribution

(also known as normal distribution) it is a bell-shaped curve.

It is assumed that during any measurement, values will follow a normal

distribution with an equal number of measurements above and below the mean value (mu).

**</summary>**

**<param** **name**=**"mu">**The mean of the distribution**</param>**

**<param** **name**=**"sigma">**The standard deviation of the distribution**</param>**

**<returns>**A data reading within a Gaussian Distribution rounded to 4 decimal points**</returns>**

**</member>**

**<member** name=**"M:Galileo.ReadData.SensorB(System.Double,System.Double)">**

**<summary>**

Sensor B records data that represents a Box-Muller distribution

(also known as normal distribution) it is a bell-shaped curve.

It is assumed that during any measurement, values will follow a normal

distribution with an equal number of measurements above and below the mean value (mu).

**</summary>**

**<param** **name**=**"mu">**The mean of the distribution**</param>**

**<param** **name**=**"sigma">**The standard deviation of the distribution**</param>**

**<returns>**A data reading within a Box-Muller Distribution rounded to 4 decimal points**</returns>**

**</member>**

**</members>**

**</doc>**