|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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 2) | | | | | | | | | | **State code:** | | | | BFF9 |
| **Assessment Title** | | Assessment Task Two Team Project | | | | | | | | | | | | | | |
| **Unit National Code & Title** | | ICTPRG440 Apply introductory programming skills in different languages | | | | | | | | | | | | | | |
| ICTPRG437 Build a user interface | | | | | | | | | | | | | | |
| ICTICT435 Create technical documentation | | | | | | | | | | | | | | |
| **Due Dates** | | Sprint One: Week Ten | | | | | | **Date Received** | | | | | | |  | |
| Sprint Two: Week Thirteen | | | | | | **Date Received** | | | | | | |  | |
| Handover: Week Fourteen | | | | | | **Date Received** | | | | | | |  | |
| **Student Name** | |  | | | | | | | | | **Student ID** | | | | |  |
| **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 | | | **Reassessment 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 | Team Project |
| Duration of Assessment | 7 Class Sessions (Week 8 - 14) |
| 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 checklist and marking guide. |
| Instructions | This project uses an agile methodology consisting of Sprint One, Sprint Two with a final submission of a Handover. Assessors need to ensure that students are in teams of two. They are required to select a Scrum Master for each sprint to lead the team and submit the documents required for each sprint (in Sprint One Question – 1 to 7 – Sprint Two Questions 8 to 14). Assessor must observe the students fulfilling their team responsibilities either as a Scrum Master or a team member using the Observation Checklist. In Sprint Two the roles will be reversed, and the Scrum Master will become the team member and vice-versa.  Students will need to demonstrate their workflow and code versioning by providing access to their GitHub account  In order to verify the authenticity of the student’s assessment, you may ask the student to again produce an answer to an existing question. |
| **TO THE STUDENT** | |
| Purpose of Assessment | You are required to show you can:  ICTPRG440 Apply introductory programming skills in different languages   1. Demonstrate your skills and knowledge by creating a GUI based application 2. Code using data structures and standard algorithms for searching and sorting data. 3. Debug, document and test completed application using IDE and associated features.   ICTPRG437 Build a user interface   1. Demonstrate your knowledge by researching prototyping tools and application development languages. 2. Investigating organizational guideline, policies and procedures.   ICTICT435 Create technical documentation   1. Demonstrate your knowledge of technical document styles and design. 2. Investigate organisational policies, procedures and standards that cover document design. 3. Document scripts for internal and external stakeholders. 4. Collaborate and discuss ideas and requirements with team members.   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.  You are required to meet the elements, performance criteria and oral communication items as outlined in the provided checklist. |
| Allowable Materials | Blackboard (Topic by topic) will include the following: Weekly Readings, Class notes, and Weekly Activities. |
| Required Resources | Computer with:   1. Web links and example code can be downloaded from the Blackboard portal 2. MS Visual Studio, 3. MSOffice 4. 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 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. |
| Project contents | This team project consists of the following tasks:   * Question 1 – Complete the Project Specification, * Question 2 – Create a GUI design, * Question 3 – Algorithm and pseudo code, * Question 4 – Scrum board and meetings, * Question 5 – Sign off and approval, * Question 6 – Develop the Sprint One application, * Question 7 – Presentation of the completed Sprint One application and associated documentation. * Question 8 – Compete the Updated Project Specification, * Question 9 – Updated GUI design, * Question 10 – Algorithm and pseudo code, * Question 11 – Scrum board and meetings, * Question 12 – Sign off and approval, * Question 13 – Develop the Sprint Two application, * Question 14 – Presentation of the completed Sprint Two application and associated documentation * Question 15 – Complete the Test Report and handover |

# Scenario

You have accepted the role of a Mid-Level Programmer for CITE Managed Services working as part of a small team to design, code and test a series of algorithms. Your task is to demonstrate your teamwork skills and programming knowledge by producing a program that will process astronomical data which is streamed from a local observatory. The continuous data represents the interaction of neutrinos with earth matter which are amplified and stored in an array data structure for analysis. The demonstration of the final application will utilise simulated data. Ref: https://en.wikipedia.org/wiki/Neutrino\_astronomy

## Software Development Methodology

This team project will use an agile methodology to complete the development of the Astronomical Processing application. For detailed information about the agile software development methodology read the PDF documents in the Software Development section on Blackboard.

It is critical that all team members are familiar with all aspects of the development. There are three major milestones, the first two will be the assessment point which focuses on the Scrum Master (Sprint One and Sprint Two). The other team member will also submit but is not required to present the Sprint application or answer interview questions. The third milestone is for the whole team and represents the final assessment point for the project (Handover). The Weekly Schedule (show below) outlines each of these milestones and the final group handover. As the development progresses the team will collect information, create mini-reports and develop the Astronomical Processing application.

## Weekly Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| MILESTONE | | TASK | DESCRIPTION |
| Week One  (week 8) | Sprint One | Question One - Five.  Analysis and Design Documentation | Complete the Project Specification documentation for the analysis and design stage.  Review the documentation and submit for Approval and Sign Off |
| Week Two  (week 9) | Sprint One | Question Six.  Application Development | The first Scrum Master will lead the development of the application using the approved Program Specifications document. |
| Week Three  (week 10) | Sprint One | Question Seven.  **Assessment Point for Scrum Master and team member.** | The first Scrum Master will present the application with Sprint One Program Criteria. The lecturer/assessor can ask questions. |
| Week Four  (week 11) | Sprint Two | Question Eight - Twelve.  Modify application | Complete the Updated Project Specification documentation for the client’s modifications.  Review the documentation and submit for Approval and Sign Off |
| Week Five  (week 12) | Sprint Two | Question Thirteen.  Modify Application | The second Scrum Master will lead the development of the application using the approved Updated Project Specifications. |
| Week Six  (week 13) | Sprint Two | Question Fourteen.  **Assessment Point for Scrum Master and team member** | The second Scrum Master will present the application with Sprint Two Program Criteria. The lecturer/assessor can ask questions. |
| Week Seven  (week 14) | Handover | Question Fifteen  **Assessment Point for all team members** | The Team conduct testing and then present the application and test report. The lecturer/assessor can ask questions of both team members. |

## Rapid Application Team

Before you can start you will be assigned membership into a team of two students. Your Lecturer will have the final decision on teams and team membership; and decide which student(s) will be allocated to each team.

The following information must be completed before the team begins work and starts the Project. Begin by having a meeting and decide on a team name, then select the Scrum Master for each Sprint, the Scrum Master must rotate for each of the two sprints. Your lecturer may select the Scrum Master or adjudicate in cases of conflict.

Team Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Sprint One

Scrum Master \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Sprint Two

Scrum Master \_\_\_\_\_\_\_\_\_\_\_\_\_

Team Member \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You should consult with the CITEMS 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 website, additional information can be collected from the Internet, ensure all sources are fully referenced. You should write your answers in the sections of the standard template provided in this document.

# Sprint One

At the local observatory the number of neutrino interactions per hour is saved as an integer value. The client wants to record and process these hourly values by storing them in an array of size 24. They require a forms-based GUI application which displays the data in a list box and uses text boxes and buttons for processing. Use the following client requirements to design a prototype of the user interface.

### Client Requirements

1. All data is stored as integers in an array.
2. The client can use a text box input to search the array.
3. There are buttons that can sort and search the data.
4. There is an input field (text box) so data can be deleted, added and edited.
5. The name of the application should be Astronomical Processing.

### Program Functionality

1. The array is of type integer.
2. The array has 24 elements to reflect the number of hours per day.
3. The sort method must be coded using the Bubble Sort algorithm.
4. The search method must be coded using the Binary Search algorithm.
5. A single text box is provided for the search input.
6. The program must generate an error message if the text box is empty.
7. The program must generate an error message if the search is not successful.
8. The program must generate a message if the search is successful.
9. The program must be able to add, edit and delete data values.
10. The array is filled with random integers to simulate the data stream (numbers between 10 and 99).

### Project Management and Source Control

During this project you will require a source control facility to save, edit and review the documents and programming code. If your do not have a GitHub\* account then you will need to create one and share the project folder with your team and provide access to your lecturer. There are several types of accounts; you should decide which type suits your requirements. GitHub also includes a Project Board option that can be used to manage the workflows (Sprints). This type of source control will be used in other SMTAFE courses.

Main Site: https://github.com/

Help Files: https://docs.github.com/en

It is the Scrum Masters responsibility to fill in the Scrum Board using the GitHub Project Board for each week. Complete the Sprint Scrum Board with information and connected team members for each week and save a screen captures of your GitHub Project Board with associated notes. This information will be presented in Question Three.

\*NOTE: GitHub is the recommended sources control, check with your lecturer if you wish to use an alternative.

### Application and GUI Prototype Design

The team should meet and discuss the Sprint One Client Requirements and Program Functionality for a design of a user interface that would be suitable for this application. The team should design, write and review the algorithm and pseudo code for the two major processing functions required by the client (Binary Search and Bubble Sort). This meeting and the associated discussions must be fully documented and added to the source control for assessment. Ensure all documentation conforms to CITEMS organizational guidelines.

Use the Sprint One Project Specifications form to complete Question One, include suitable information for each of the sections. Add your meeting notes, Scrum Board screen captures (etc) to the end of this document.

## Question One

### Instructions

Fill in all sections of the Sprint One Project Specifications form, start by completing the Project Details. Then list all the Project Tasks and assign a Priority from Very Important = 1 to Least Important = 5. Next, list all the functional and non-functional requirements based on the client requirements and program functionality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint One Project Specifications | | | | |
| Project Details | | | | |
| Release | | 1.0 | | |
| Date | |  | | |
| Team Name | |  | | |
| Scrum Master | |  | | |
| Team Member | |  | | |
| Project Tasks | | | | |
| Task # | Description | | Priority | Notes |
| 1 |  | |  |  |
| 2 |  | |  |  |
| 3 |  | |  |  |
| 4 |  | |  |  |
| 5 |  | |  |  |
| 6 |  | |  |  |
|  |  | |  |  |
|  |  | |  |  |
| Functional Requirements | | | | |
| A functional requirement is **describing the behaviour of the system** as it relates to the system's functionality. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
| Non-Functional Requirements | | | | |
| A non-functional requirement elaborates a performance characteristic of the system. | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |
|  | | | | |

## Question Two

### Instructions

Fill in all sections of the GUI Design Specifications form. List all the UI components (buttons, textbox, listbox, etc) and describe what action or event is associated with each. Insert a detailed image/picture/screen capture of the GUI design, ensure all the features are identified and labelled.

|  |  |
| --- | --- |
| GUI Design Specifications | |
| GUI Development Tool |  |
| Application Development IDE |  |
| UI Components | |
| List all UI components and their associated action/event (describe what each component does). | |
| GUI Layout | |
| Insert Your Diagram/Image here | |

## Question Three

### Instructions

Fill in the two sections for the algorithm and pseudo code for the Binary Search and Bubble Sort. Refer to the Blackboard resources to ensure your answer reflects the differences between an Algorithm and Pseudo Code.

|  |  |  |
| --- | --- | --- |
| Algorithm/Pseudo Code Design for Binary Search | | |
| Put the Algorithm for the Binary Search here; | Put the Pseudo Code for the Binary Search here; | |
| Algorithm/Pseudo Code Design for Bubble Sort | | |
| Put the Algorithm for the Bubble Sort here; | | Put the Pseudo Code for the Bubble Sort here; |

## Question Four

### Instructions

Fill in the two sections for the Scrum Board snapshot and meeting agenda/minutes. Ensure these documents are reflected in the Source Control.

|  |  |
| --- | --- |
| Scrum Board and Meeting Notes | |
| Put the Scrum Board and meeting notes here; |  |

## Question Five

### Sprint

### One Approval

The Scrum Master will arrange for the completed Sprint One Project Specification document to be reviewed by the Lecturer/Assessor for approval, sign off and feedback before the team starts Question Six.

Your submission of the Sprint One Project Specification will include:

* Question One
  + Project Details,
  + Project Tasks,
  + Requirements (Functional and non-functional),
* Question Two
  + Design Specifications (detailed diagram of the GUI design),
  + GUI Layout,
* Question Three
  + Algorithm and pseudo code for the Binary Search and Bubble Sort.
* Question Four
  + Scrum Board and meeting details (screen shots from source control)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
|  |  |  |  |  |
|  |  |  |  |  |
| Lecturer Feedback | | | | |
|  | | | | |
| Meeting, discussion, and related notes | | | | |
|  | | | | |

## Question Six

### Programming and Application Development

Once the team has received approval from the Lecturer/Assessor review and amend the Sprint One Project Specification document based on the feedback. The Scrum Master should complete the Scrum Board for Week Two and the team should begin to create a Windows Forms Prototype Application that will satisfy the Client Requirements and Program Functionality.

As the team progresses and completes tasks the Scrum Master will update the relevant Scrum Board. Avoid deleting items, simply add new items into the next column, this will provide historical evidence of the teams’ progress.

The Windows Form Prototype Application must comply with CITEMS organisational requirements for coding standards, comments, and documentation. For example, add suitable comments to all your code. Add a header comment at the top of the code as shown below, with Name(s), Date, Version, and a program description.

**// Your Name, Team Name, Sprint Number**

**// Date:**

**// Version:**

**// Name of the program**

**// Brief explanation of the program and list,**

**// Inputs, Processes, Outputs**

## Question Seven

### Presentation and Sprint One Review

The Scrum Master should update the week three Scrum Board and ensure the programming code has the correct naming conventions and internal comments as per the CITEMS Policies and Standards. Notify your Lecturer to arrange a suitable time to present the final Sprint Two documents and demonstrate the working Astronomical Processing application. This presentation will be assessing the Scrum Master using simulated data in the array. The following Submission Requirements should be used to ensure all aspects of the assessment are covered.

## Submission Requirements

Your submission for Sprint One will include:

Completed Project Specifications form with Lecturer/Assessor approval from Question One,

Completed Meeting and Discussion notes,

Completed Scrum Board document (3 completed scrum boards),

Completed Solution Folder for the Astronomical Processing application

The Sprint One Scrum Master will demonstrate the program which must satisfy the following:

* The program must load random data into the array using a button click.
* The client must be able to click a button to bubble sort the data.
* The client must be able to enter search data and click a button to activate a binary search.
* All data is displayed in a ListBox.
* Data can be added, edited and deleted.
* All error messages are demonstrated.
* Code comments are relevant.
* Demonstrate your workflow and code versioning within GitHub.
* Answer questions on all aspects of the Sprint One development and documents.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint One  Assessment Task Two  Marking Guide and Observation Checklist | | | | |
| Student Name  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Tick one | | | |
| Scrum Master | | | Team Member |
| Demonstration Criteria | Satisfactory | | Comment | |
| 1. The program must load random data into the array using a button click. | Yes | No |  | |
| 1. The client must be able to click a button to bubble sort the data. | Yes | No |  | |
| 1. The client must be able to enter search data and click a button to activate a binary search. | Yes | No |  | |
| 1. All data is displayed in a ListBox | Yes | No |  | |
| 1. Data can be added, edited and deleted | Yes | No |  | |
| 1. All user input in team meetings messages are demonstrated   ie Incorrect integer input (ie “A”, char) | Yes | No |  | |
| 1. Code comments are relevant. | Yes | No |  | |
| 1. Three Completed Scrum Boards on GitHub and the following has been checked:    * history of each Project Board    * design has been checked to see that the code has been developed, designed, reviewed and maintained over time. | Yes | No |  | |
| 1. Project Specification completed as per the organisational template provided. | Yes | No |  | |
| Observation Checklist | Satisfactory | | Comment | |
| 1. Participated fully in team discussion/meetings in all aspects of the development process | Yes | No |  | |
| 1. Followed the appropriate software development standards and workflow when creating all code (using an IDE to write, document, test, and debug) | Yes | No |  | |
| 1. Create and share GitHub resources with the team – assessor to ensure GitHub versioning and history is checked. | Yes | No |  | |
| 1. Asked appropriate questions in relation to all areas of design and development | Yes | No |  | |
| 1. Listens and responds accordingly to team members | Yes | No |  | |
| 1. Speaks clearly using technical language ensuring that the project progresses. | Yes | No |  | |
| 1. Discusses critical aspects of technical specifications providing alternative perspectives to the team | Yes | No |  | |
| 1. Collaborated in the Project Specifications document | Yes | No |  | |
| 1. Collaborated with coding and app development confirmed in GitHub versioning and history | Yes | No |  | |
| 1. Uses and develops standard algorithms and MSDN language standards | Yes | No |  | |
| 1. Applies modular programming principles separating the functionality of a program into independent, interchangeable modules | Yes | No |  | |
| 1. Completes designated tasks from Project Board – confirmed on GitHub | Yes | No |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessor Name** |  | | | | |
| **Assessment Decision** |  | Satisfactory |  | Not Yet Satisfactory | |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | **Reassessment Date:** | |  |
| **Assessor Signature** |  | | **Date** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
|  | | | |
| **Student signature** |  | **Date** |  |

End of Sprint One

## Software Development (revisited)

This team project will use an agile methodology to complete the development of the Astronomical Processing application. For detailed information about these software development methodologies read the PDF documents in the Software Development section on Blackboard.

It is critical that all team members are familiar with all aspects of the development. There are three major milestones, the first two will be the assessment point which focuses on the Scrum Master (Sprint One and Sprint Two). The other team member will also submit but is not required to present the Sprint application or answer questions. The third milestone is for the whole team and represents the final assessment point for the project (Handover). The Updated Weekly Schedule (show below) outlines each of these milestones and the final group handover. As the development progresses the team will collect information, create mini-reports and develop the Astronomical Processing application.

## Updated Weekly Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| MILESTONE | | TASK | DESCRIPTION |
| Week One  (week 8) | Sprint One | Completed |  |
| Week Two  (week 9) | Sprint One | Completed |  |
| Week Three  (week 10) | Sprint One | Completed |  |
| Week Four  (week 11) | Sprint Two | Question Eight - Twelve.  Modify application | Complete the Updated Project Specification documentation for the client’s modifications.  Review the documentation and submit for Approval and Sign Off |
| Week Five  (week 12) | Sprint Two | Question Thirteen.  Modify Application | The second Scrum Master will lead the development of the application using the approved Updated Project Specifications. |
| Week Six  (week 13) | Sprint Two | Question Fourteen.  **Assessment Point for Scrum Master and team member** | The second Scrum Master will present the application with Sprint Two Program Criteria. The lecturer/assessor can ask questions. |
| Week Seven  (week 14) | Handover | Question Fifteen  **Assessment Point for all team members** | The Team conduct testing and then present the application and test report. The lecturer/assessor can ask questions of both team members. |

## Rapid Application Team

In Sprint Two the roles of the two team members are reversed, therefore the previous Scrum Master will take on the role of team member.

You should consult with the CITEMS 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 website, additional information can be collected from the Internet, ensure all sources are fully referenced. You should write your answers in one of the standard templates provided on Blackboard.

# Sprint Two

Following the success of Sprint One the client has requested several modifications and additions to the Astronomical Processing application. The client would like several additional buttons to perform mathematical calculations on the data set. Furthermore, the client would like to add a sequential search option. Use the following client requirements and program criteria to amend the application.

### Client Requirements

1. A button to calculate the mid-extreme,
2. A button to calculate the mode,
3. A button to calculate the average,
4. A button to calculate the range.
5. All GUI components have detailed tool tips.
6. A button to initiate a sequential search.

### Program Functionality

1. The mathematic calculations will display in separate text boxes formatted to 2 decimal places as appropriate.
2. The sequential sort method must be coded using a single FOR loop and one IF condition.
3. The program must generate an error message if the text box is empty.
4. The program must generate an error message if the search is not successful.
5. The program must generate a message if the search is successful.

### Project Management and Source Control

During this project you are required to use a source control facility to save, edit and review the documents and programming code.

It is the Scrum Masters responsibility to fill in the Scrum Board using the GitHub Project Board for each week. Complete the Sprint Scrum Board with information and connected team members for each week and save screen captures of your GitHub Project Board with associated notes. This information will be presented in Question Six.

### Application and GUI Specification Update

The team should meet to discuss and plan the Sprint Two Client Requirements and Program Functionality. The team should design and write the algorithm and pseudo code for the sequential search method required by the client. Finally, the team should develop suitable algorithms and pseudo code for each of the four mathematical functions. This meeting and the associated discussions must be fully documented and added to the source control for assessment. Ensure all documentation conforms to CITEMS organizational guidelines.

Use the following Sprint Two Project Specifications form to complete Question Four, include suitable information for each of the sections. Add your meeting notes, Scrum Boards screen captures (etc) to the end of this document.

## Question Eight

### Instructions

Fill in all sections of the Sprint Two Project Specifications form, start by completing the Project Details. Then list all the Project Tasks and assign a Priority from Very Important = 1 to Least Important = 5. Next, list all the functional and non-functional requirements based on the client requirements and program functionality.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint Two Project Specifications | | | | |
| Project Details | | | | |
| Release | | V5 | | |
| Date | | 6/10/2021 | | |
| Team Name | | AJ | | |
| Scrum Master | | Jesse Newill | | |
| Team Member | | Alexis Culley | | |
| Project Tasks | | | | |
| Task Name | Description | | Priority | Notes |
| 1 | Team meeting 1 to assign roles and discuss project requirements. Project broken down into its components and tasks assigned. | | 1 | Both |
| 2 | Scrum board creation. | | 1 | Jesse |
| 3 | Identify and document functional requirements | | 1 | Alexis |
| 4 | Identify and document Non-Functional requirements | | 1 | Jesse |
| 5 | Draft design of GUI and note UI components | | 1 | Both |
| 6 | Complete pseudo code for Sequential Search | | 2 | Jesse |
| 7 | Complete pseudo code for Mode | | 2 | Alexis |
| 8 | Complete pseudo code for Mid - extreme | | 2 | Jesse |
| 9 | Complete pseudo code for Average | | 2 | Jesse |
| 10 | Complete pseudo code for Range | | 2 | Alexis |
| 11 | Document notes for team meeting 1 | | 1 | Alexis |
| 12 | Review the pseudo code | | 2 | Both |
| 13 | Team meeting 2 to review and edit all work prior to seeking approval to proceed to question 13 | | 2 | Both |
| 14 | Document notes for team meeting 2 | | 2 | Alexis |
| 15 | A button to calculate the mid-extreme | | 3 | Jesse |
| 16 | A button to calculate the mode | | 3 | Jesse |
| 17 | A button to calculate the range. | | 3 | Jesse |
| 18 | All GUI components have detailed tool tips. | | 3 | Jesse |
| 19 | A button to initiate a sequential search. | | 3 | Jesse |
| 20 | A button to calculate the average | | 3 | Jesse |
| 21 | Debugging code | | 4 | Jesse |
| 22 | Team meeting 3 to review feedback and make any necessary changes | | 3 | Both |
| 23 | Document notes for team meeting 3 | | 3 | Alexis |
| 24 | Testing | | 4 | Alexis |
| 25 | Test Reporting | | 4 | Alexis |
| 26 | Mapping | | 4 | Alexis |
| 27 | Commenting | | 4 | Jesse |
| 28 | Team meeting 4 to review prototype application and remediate any issues | | 5 | Both |
| 29 | Document notes for team meeting 4 | | 5 | Alexis |
| 30 | Prepare, deliver, and upload to Blackboard | | 5 | Jesse |
| Functional Requirements | | | | |
| The functional requirement is **describing the behaviour of the system** as it relates to the system's functionality. | | | | |
| Original Requirements:   * All data should be stored in arrays as integers * The user of the program should be able to search for data in the array, using the text box * Buttons are in the user interface for the purpose of sorting and searching data * Data should be able to be added, deleted, and edited from the text box * The application’s name should be Astronomical Processing * They array should have twenty-four elements to represent the hours in a day * The search input should be done within a textbox * There must be an error message generated if there is nothing the textbox * There must be an error message generated if the search was not successful * There must be a message generated for a successful search * Values must be able to be added, deleted, and edited. * The array must also be able to simulate the data stream with random numbers   Additional Requirements:   * A button to calculate the mid extreme. * A button to calculate the mode. * A button to calculate the range. * A button to calculate the average. * A button to initiate a sequential search option. * The program must generate an error message if the text box is empty. * The program must generate an error message if the search is not successful. * The program must generate a message if the search is successful. * All GUI components have detailed tool tips. * The mathematic calculations will display in separate text boxes formatted to 2 decimal places as appropriate. | | | | |
| Non-Functional Requirements | | | | |
| The non-functional requirement elaborates a performance characteristic of the system. | | | | |
| Original Requirements:   * Binary search algorithm for search type. * Data being stored in an array. * Integer data type for input. * Bubble sort algorithm for sort type.   Additional Requirements:   * Sequential Search should be used * Integer data type for input * Data being stored in an array * The sequential sort method must be coded using a single FOR loop and one IF condition | | | | |

## Question Nine

### Instructions

Fill in all sections of the GUI Design Specifications form. List all the new UI components (buttons, textbox, listbox, etc) and describe what action or event is associated with each. Insert a detailed image/picture/screen capture of the updated GUI design, ensure all the features are identified and labelled.

|  |  |
| --- | --- |
| GUI Design Specifications | |
| GUI Development Tool | Adobe XD |
| Application Development IDE | Microsoft Visual Studio |
| UI Components | |
| List all UI components and their associated action/event (describe what each component does).   * Mid Extreme Button: Calculates the mid extreme. * Mode Button: Calculates the mode. * Range Button: Calculates the range. * Sequential Search Button: Initiates a sequential search. * Add Button: Used to add data to the list box. * Edit Button: Used to edit data displayed in the list box. * Delete Button: Used to delete data displayed in the list box. * Sort Button: Used to sort data displayed in the list box. * Search Button: Used to search data displayed in the list box. * Textbox: Field that accepts user input to display in the list box. * Status Strip: Displays the system status. * Message Box: Displays error messages. * Auto Fill Button: Populates array with random data for testing purposes. * Test Box 1: Displays the data input by the user. * Test Box 2: Displays the mid-extreme. * Test Box 3: Displays the mode. * Test Box 4: Displays the average. * Test Box 5: Displays the range. | |
| GUI Layout | |
| Insert Your Diagram/Image here  Graphical user interface, application  Description automatically generated | |

## Question Ten

### Instructions

Fill in the two sections for the algorithm and pseudo code for the Binary Search and Bubble Sort. Refer to the Blackboard resources to ensure your answer reflects the differences between an Algorithm and Pseudo Code.

The client would like the following definitions to be used when developing the code for the four mathematical functions:

1. Mid-Extreme: The mid-extreme is defined as the sum of the smallest value and the largest value in the given data set divided by 2.
2. Mode: The mode is defined as the number that appears most frequently in a set of data (unimodal).
3. Average: The average is defined as the sum of all the values divided by the total number of values in the data set.
4. Range: The range is defined as the difference between the largest and the smallest values in the data set.

|  |  |
| --- | --- |
| Algorithm/Pseudo Code Design for Sequential Search | |
| Put the Algorithm for the Sequential Search here;  A Sequential search looks at all the items in a list and will go through all of them until it finds the one it matches with. If nothing matches with it, then the search will fail. | Put the Pseudo Code for the Sequential Search here;  try  for each item in the list  if match item == value  return the item's location  end if  end for  catch  display error  if (not found)  return item not found | |
| Algorithm/Pseudo Code Design for Mathematical Functions | |
| Put the Algorithm for the Math Functions here;  **MODE:**  The number that appears most frequently in a set of data.   1. The array is searched to determine which number appears most frequently in the array. 2. The system returns the number that appears most frequently and the number of times it appears.   **MID-EXTREME:**  The sum of the smallest value and the largest value in a data set, divided by two.   1. The minimum and maximum values are assigned and compared to the numbers in the array. 2. The system returns the minimum and maximum numbers in the array. 3. The two numbers are added together and divided by two.   **AVERAGE:**  The average is found by talking the sum of all the numbers in a set and then dividing the sum by the amount of numbers in that set.  **RANGE:**  A range in found by taking the difference of the largest number and the smallest number. | Put the Pseudo Code for the Math Functions here;  **MODE:**  for (integer i = 0; i < array length; i++)  counter = 0  element = array[i]  for(integer j = 0; j < array length; j++)  if element == array[j]  counter++  if(counter >= frequency)  frequency = counter  mode = element  Return element value and frequency  **MID-EXTREME:**  minValue = 100  maxValue = 0  for (x = 0, x = length of the array, x++)  if array x < = minValue  minValue = ary[x]  if array[x] >=maxValue  maxValue = ary[x]  Return maxValue + minValue / 2    **AVERAGE:**  double sum=0  double avg=0              double array numbers              for(int i=0, I < length of numbers, i++)                  sum += numbers[i]                avg = sum / length of numbers  **RANGE:**  **Input:** arr[] = {15, 16, 10, 9, 6, 7, 17}  **Output:** Range : 11  Max = 17, Min = 6  Range = Max – Min = 17 – 6 = 11 | |

## Question Eleven

### Instructions

Fill in the two sections for the Scrum Board snapshot and meeting agenda/minutes. Ensure these documents are reflected in the Source Control.

|  |  |
| --- | --- |
| Scrum Board and Meeting Notes | |
| Put the Scrum Board and meeting notes here;  Graphical user interface, application  Description automatically generatedScrum board as of 6/10/2021:  Scrum board as of 13/10/2021:  Graphical user interface, application  Description automatically generated  Scrum board as of 20/10/2021:  Graphical user interface, application  Description automatically generated | Team Meeting 1 Notes 06/10/21:    Team Meeting 2 Notes 06/10/21:    Team Meeting 3 Notes 06/10/21:    Team Meeting 4 Notes |

## Question Twelve

### Sprint Two Approval

The Scrum Master will arrange for the completed Sprint Two Project Specification document to be reviewed by the Lecturer/Assessor for approval, sign off and feedback before the team starts Question Thirteen.

Your submission of the Sprint Two Project Specification will include:

* Question Eight
  + Project Details,
  + Project Tasks,
  + Requirements (Functional and non-functional),
* Question Nine
  + Design Specifications (detailed diagram of the GUI design),
  + GUI Layout,
* Question Ten
  + Algorithm and pseudo code for the Sequential Search and four Mathematical Functions.
* Question Eleven
  + Scrum Board and meeting details (screen shots from source control)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approval (Lecturer/Assessor use only) | | | | |
| Approver Name | Title | Signature | Date | Approved? |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Feedback and General Notes | | | | |
|  | | | | |
| Meeting, discussion, and related notes | | | | |
|  | | | | |

## Question Thirteen

### Programming and Application Modification

Once your team has received approval from the Lecturer/Assessor review and amend the Sprint Two Project Specification document based on the feedback. The Scrum Master should complete the Scrum Board for Week Five and the team should begin work on the modification for the Astronomical Processing application to satisfy the Client Requirements and Program Functionality.

As the team progresses and completes tasks the Scrum Master will update the relevant Scrum Board. Avoid deleting items, simply add new items into the next column, this will provide historical evidence of the teams’ progress.

The Windows Form Prototype Application must comply with CITEMS organisational requirements for coding standards, comments, and documentation.

## Question Fourteen

### Presentation and Sprint Two Review

The Scrum Master should update the week six Scrum Board and ensure the programming code has the correct naming conventions and internal comments as per the CITEMS Policies and Standards. Notify your Lecturer to arrange a suitable time to present the final Sprint Two documents (Scrum Boards and Project Specifications) and the working Astronomical Processing application. This presentation will be assessing the Scrum Master using simulated data in the array. The following Marking Guide should be used to ensure all aspects of the assessment are covered.

## Submission Requirements

Your submission for Sprint Two will include:

Complete Project Specifications documentation with Lecturer/Assessor approval from Question Four,

Completed Meeting and Discussion notes,

Completed Scrum Board document (3 completed scrum boards),

Complete Solution Folder for the Astronomical Processing application

The Sprint Two Scrum Master will demonstrate the program which must satisfy the following:

1. The program must load random data into the array using a button click.
2. The client must be able to enter search data and click a button to activate a linear search.
3. The client can click each function button which will populate the related textbox.
4. The mathematical results are properly formatted.
5. All major components have detailed tool tips.
6. All error messages are demonstrated.
7. Code comments are relevant.

* Answer questions on all aspects of the Sprint Two development and documents.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint Two  Assessment Task Two  Marking Guide and Observation Checklist | | | | |
| Student Name  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Tick one | | | |
| Scrum Master | | | Team Member |
| Demonstration Criteria | Satisfactory | | Comment | |
| * The program must load random data into the array using a button click. | Yes | No |  | |
| * The client must be able to enter search data and click a button to activate a linear search. | Yes | No |  | |
| * The client can click each function button which will populate the related textbox. | Yes | No |  | |
| * The mathematical results are properly formatted. | Yes | No |  | |
| * All major components have detailed tool tips. | Yes | No |  | |
| * All user input messages are demonstrated. i.e. Incorrect integer input (ie “A”, char) | Yes | No |  | |
| * Code comments are relevant. | Yes | No |  | |
| * Three Completed Scrum Boards   + history of each Project Board   + designed has been checked to see that the code has been developed, designed, reviewed and maintained over time. | Yes | No |  | |
| * Project Specifications completed as per the organisational template provided. | Yes | No |  | |
| Observation Checklist | Satisfactory | | Comment | |
| * Participated fully in team discussion/meeting in all aspects of the development process | Yes | No |  | |
| * Followed the appropriate software development standards and workflow when creating all code (using an IDE to write, document, test, and debug) | Yes | No |  | |
| * Create and share GitHub resources - assessor to ensure GitHub versioning and history is checked. | Yes | No |  | |
| * Asked appropriate questions in relation to all areas of design and development | Yes | No |  | |
| * Listens and responds accordingly to team members | Yes | No |  | |
| * Speaks clearly using technical language ensuring that the project progresses. | Yes | No |  | |
| * Discusses critically aspects of technical specifications | Yes | No |  | |
| * Collaborated in the Project Specifications document | Yes | No |  | |
| * Collaborated with coding and app development as confirmed on GitHub | Yes | No |  | |
| * Uses and develops standard algorithms and MSDN language standards | Yes | No |  | |
| * Applies modular programming principles separating the functionality of a program into independent, interchangeable modules | Yes | No |  | |
| * Completes designated tasks from Project Board as confirmed on GitHub | Yes | No |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessor Name** |  | | | | |
| **Assessment Decision** |  | Satisfactory |  | Not Yet Satisfactory | |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | **Reassessment Date:** | |  |
| **Assessor Signature** |  | | **Date** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
|  | | | |
| **Student signature** |  | **Date** |  |

End of Sprint Two

## Software Development (Handover)

This team project will use an agile methodology to complete the development of the Astronomical Processing application. For detailed information about the agile software development methodology read the PDF documents in the Software Development section on Blackboard.

It is critical that all team members are familiar with all aspects of the development. There are three major milestones, the first two will be the assessment point which focuses on the Scrum Master (Sprint One and Sprint Two). The other team member will also submit but is not required to present the Sprint application or answer questions. The third milestone is for the whole team and represents the final assessment point for the project (Handover). The Weekly Schedule (show below) outlines each of these milestones and the final group handover. As the development progresses the team will collect information, create mini-reports and develop the Astronomical Processing application.

## Weekly Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| MILESTONE | | TASK | DESCRIPTION |
| Week One  (week 8) | Sprint One | Completed |  |
| Week Two  (week 9) | Sprint One | Completed |  |
| Week Three  (week 10) | Sprint One | Completed |  |
| Week Four  (week 11) | Sprint Two | Completed |  |
| Week Five  (week 12) | Sprint Two | Completed |  |
| Week Six  (week 13) | Sprint Two | Completed |  |
| Week Seven  (week 14) | Handover | Question Fifteen  **Assessment Point for all team members** | The Team conduct testing and then present the application and test report. The lecturer/assessor can ask questions of both team members. |

## Rapid Application Team

In Sprint One and Sprint Two each team member assumed the role of Scrum Master and Team Member; in this final task all team members are equal and share the responsibility to complete all the assessment criteria.

You should consult with the CITEMS 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 website, additional information can be collected from the Internet, ensure all sources are fully referenced. You should write your answers in one of the standard templates provided on Blackboard.

# Handover

The client has accepted the sprint one and sprint two development milestones for the Astronomical Processing application. The final stage is to conduct testing on all methods that calculate the four mathematic functions. This is to validate the accuracy of the methods.

The client wants the tests recorded in a formal Test Report with associated screen captures of the debug session which traces the changing values for each of the four functions. Use the following client requirements and complete the Test Report by recording the testing of the four mathematical methods.

### Client Requirements

1. All four mathematical methods are tested (mid-extreme, mode, average and range).
2. The results of the tests are recorded in the formal Test Report.
3. Each mathematical function is tested more than three times.
4. Each mathematical method has a break point.
5. Each mathematical method has local variables displayed in a watch.

### Testing

Read the Client Requirements and ensure all relevant information is included in the Test Report. This is a team effort so the team should meet and discuss how each of the four mathematical functions can be fully tested. The meeting should consider what type of data will be required for each mathematical function and reflect the organisational guidelines of CITEMS (refer www.citems.com.au/). Use the Test Report during the test session, add additional rows as required and include suitable screen captures to support each test case. Where the testing highlights an issue in the Astronomical Processing application you can updated the code and record this change to ensure all client requirements and testing are satisfactory.

## Question Fifteen

### Instructions

Fill in all sections of the Test Report, start by completing the Project Details. Then list all the Test Cases and the associated Test Steps. Run the tests and record the results. Review the results and modify the code to ensure correct functionality of the application.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Report | | | | | | | | |
| Project Details | | | | | | | | |
| Date | |  | | | | | | |
| Team Name | |  | | | | | | |
| Team Members | |  | | |  | | | |
| Test Type | | **Description of the test type.** | | | | | | |
| Test Case # | Test Case Name | | Test Steps | Test Data | | Expected Results | Evidence  (ref to Screen Capture) | Pass / Fail |
| 1 |  | | **List the steps required to perform the test.** | **List the test data** | |  |  |  |
|  |  | |  |  | |  |  |  |
|  |  | |  |  | |  |  |  |
|  |  | |  |  | |  |  |  |

## Submission Requirements

Your submission for the Handover will include:

Completed Test Report from Question Fifteen,

Completed Solution Folder for the Astronomical Processing application.

Your team will submit the working program and Test Report documentation to the appropriate Blackboard section. The following Marking Guide should be used to ensure all aspects of the final Handover assessment are covered. Consult your lecturer for further information or clarification.

|  |  |  |  |
| --- | --- | --- | --- |
| Assessment Task Two Handover  Marking Guide | | | |
| Student Name | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Criteria | Satisfactory | | Comment |
| * The test data is suitable for each test case and reflects the design and development requirements. | Yes | No |  |
| * The test results are support with screen captures. | Yes | No |  |
| * The Test Report is complete. | Yes | No |  |
| * All documentation is properly formatted | Yes | No |  |
| * Participated in both sprints | Yes | No |  |
| * Participated in the testing/handover | Yes | No |  |
| * Active and supportive team members | Yes | No |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Assessor Name** |  | | | | |
| **Assessment Decision** |  | Satisfactory |  | Not Yet Satisfactory | |
| **Is student eligible for reassessment (Re-sit)?** | No | Yes | **Reassessment Date:** | |  |
| **Assessor Signature** |  | | **Date** | |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Feedback to student** | | | |
|  | | | |
| **Student signature** |  | **Date** |  |

**NOTE to Assessor: This is an individual submission. Please complete one checklist per student.**

End of Assessment