|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Qualification details** | | | | | | | | | | | | | | | | |
| **Training Package Code and Title:** | | ICT - Information and Communications Technology (Release 8.0) | | | | | | | | | | | | | | |
| **Qualification National Code and Title:** | | ICT40120 Certificate IV in Information Technology  (Release 4) | | | | | | | | | | **State code:** | | | | BFF9 |
| **Assessment Title** | | Assessment Task Three (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 14 | | | | | | **Date Received** | | | | | | |  | |
| Sprint Two: Week 17 | | | | | | **Date Received** | | | | | | |  | |
| Handover: Week 19 | | | | | | **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 | 8 Class Sessions (Week 12 - 19) |
| 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, 2, 3…7 – Sprint Two - Question 4, 5, 6…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   * Demonstrate your skills and knowledge by creating a GUI based application * Code using data structures and standard algorithms for searching and sorting data. * Debug, document and test completed application using IDE and associated features.   ICTPRG437 Build a user interface   * Demonstrate your knowledge by researching prototyping tools and application development languages. * Investigating organizational guideline, policies and procedures.   ICTICT435 Create technical documentation   * Demonstrate your knowledge of technical document styles and design. * Investigate organisational policies, procedures and standards that cover document design. * Document scripts for internal and external stakeholders. * 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:   * Web links and example code can be downloaded from the Blackboard portal * 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 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:  Sprint One   * 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.   Sprint Two   * 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   Handover   * Question 15 – Complete the Test Report * Question 16 – Recommendations * Question 17 – Review 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 12  AM session | 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. |
| PM Session | Sprint One | Question Six.  Commence Application Development | The first Scrum Master will lead the development of the application using the approved Program Specifications document. |
| Week 13  AM session | Sprint One | Question Six.  Continue Application Development | The Team will continue the development of the application using the approved Program Specifications document. |
| PM Session | Sprint One | Question Six.  Complete Application Development | The Team will complete the development of the application and test the error trapping and user functionality. |
| Week 14  AM session | Sprint One | Question Seven.  **Finalise Sprint One** | The first Scrum Master will present the application with Sprint One Program Criteria. The lecturer/assessor can ask questions. |
| PM Session | Sprint One | **Assessment Point for first 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 15  AM session | Sprint Two |  |  |
| PM Session | Sprint Two |  |  |
| Week 16  AM session | Sprint Two |  |  |
| PM Session | Sprint Two |  |  |
| Week 17  AM session | Sprint Two |  |  |
| PM session | Sprint Two |  |  |
| Week 18  AM session | Handover |  |  |
| PM Session | Handover |  |  |
| Week 19  AM session | Handover |  |  |
| PM Session | Handover |  |  |

## 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 \_\_\_BigBooleans

### Sprint One

Scrum Master \_\_\_Peter Halligan

Team Member \_\_\_ Silas Huege De Serville

### Sprint Two

Scrum Master \_\_\_ Silas Huege De Serville

Team Member \_\_\_ Peter Halligan

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

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

### Program Functionality

* The array is of type integer.
* The array has 24 elements to reflect the number of hours per day.
* The sort method must be coded using the Bubble Sort algorithm.
* The search method must be coded using the Binary Search algorithm.
* A single text box is provided for the search input.
* 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.
* The program must be able to edit data values.
* 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 | | Week 19 | | |
| Date | | 20/10/2022 | | |
| Team Name | | BigBooleans | | |
| Scrum Master | | Peter Halligan | | |
| Team Member | | Silas Huege De Serville | | |
| Project Tasks | | | | |
| Task # | Description | | Priority | Notes |
| 1 | Github, team set up | | 1 | Initial team Github set up |
| 2 | Project Spec Form | | 1 | Sort task order |
| 3 | UI Design | | 1 | Initial program Prototype |
| 4 | Bin search/Bubble sort Algorithms/Pseudo Code | | 2 | Hand out a task each |
| 5 | CITEMS Rep approval and Notes | | 3 | Amendment round |
| 6 | Question 6 First run | | 3 | Bulk code |
| 7 | Question 6 Second Run | | 4 | More code then polish |
| 8 | Presentation and Review | | 5 |  |
| Functional Requirements | | | | |
| A functional requirement is **describing the behaviour of the system** as it relates to the system's functionality. | | | | |
| 24 elements as Data type int filled as random in Array | | | | |
| Sort method using bubble sort algorithm | | | | |
| Search method using Binary search | | | | |
| Single text box input | | | | |
| Error handling for text box empty and search not successful | | | | |
| Able to edit array data | | | | |
| Non-Functional Requirements | | | | |
| A non-functional requirement elaborates a performance characteristic of the system. | | | | |
| Bubble sort must complete within one second | | | | |
| Binary search must be complete within one second | | | | |
| Error handling for array out of bounds | | | | |
| Error handling for element does not exist in array | | | | |
| Error handling for duplicate entries | | | | |

## 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).  List box, Label for textbox, Textbox, search button, status strip | |
| GUI Layout | |
|  | |

## 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; SET bool to binary search function  check if user input number of the the selected index  displays the index  SET int minIndex equal to 0  SET int maxIndex equal to array length minus 1  SET index equal to minus 1  WHILE minIndex is less than or equal to maxIndex      Index equal to minIndex plus maxIndex divide by 2      IF selected index is equal to myArray of the index          RETURN true      ELSE IF selected index is less than myArray index          maxIndex equal to index minus 1      ELSE          minIndex equal to index plus 1  RETURN false | |
| Algorithm/Pseudo Code Design for Bubble Sort | | |
| Put the Algorithm for the Bubble Sort here; | | SET temp variable as zeroforeach item in array                foreach item in array                                set left as zero                                if left is less then array length                                left plus one                                IF array index\_left is greater then array\_left plus one                                                set temp variable as array index\_left plus one                                                set array\_left plus one as array\_left                                                set array\_left as temp variable |

## 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;  2.30pm – 3:00  Decide on best way to distribute workload.  Discuss Project tasks then decide their priority and order of completion.  3:00 – 3:45  List all client requirements. Decide which are functional and which are non-functional.  4:00 – 5:00  Peter – Design Form GUI based of requirements. Create Bubble sort algorithm and pseudo code.  Silas – Cover GUI Design Specifications and create Binary Search algorithm and pseudo code |  |

## 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 for approval 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? |
| Blake Williams | Lecturer | B\_williams | 27/10/2022 | NO |
| Blake Williams | Lecturer | B\_williams | 27/10/2022 | YES |
| Lecturer Feedback | | | | |
| Your UI has too many buttons. Your bubble sort variables need renaming.  You need additional meeting notes. | | | | |
| 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 edited.
* 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 Three  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 edited | 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:    1. history of each Project Board    2. 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