|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Qualification details** | | | | | | | | | | | | | | | | |
| **Training Package Code and Title:** | | ICT - Information and Communications Technology (Release 8.1) | | | | | | | | | | | | | | |
| **Qualification National Code and Title:** | | ICT40120 Certificate IV in Information Technology  (Release 4) | | | | | | | | | | **State code:** | | | | BFF9 |
| **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:  YES  ………………………………………….. | | | | | | | | | | | | | | |
| **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.

The team will utilise all the skills from previous assessment tasks and the example code and documentation on Blackboard.

Updated Weekly Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| MILESTONE | | TASK | DESCRIPTION |
| Week 12  AM session | Sprint One |  |  |
| PM Session | Sprint One |  |  |
| Week 13  AM session | Sprint One |  |  |
| PM Session | Sprint One |  |  |
| Week 14  AM session | Sprint One |  |  |
| PM Session | Sprint One |  |  |
| Week 15  AM session | Sprint Two | Question Eight - Twelve.  Analysis and Design Documentation | Review the Updated Project Specification documentation for the client’s modifications.  Review the documentation and submit for Approval and Sign Off. |
| PM Session | Sprint Two | Question Thirteen.  Commence Development to Modify Application | The second Scrum Master will lead the development of the application using the approved Updated Project Specifications. |
| Week 16  AM session | Sprint Two | Question Thirteen.  Continue Development to Modify Application | The Team will continue the development of the application with Sprint Two Program Criteria. |
| PM Session | Sprint Two | Question Thirteen.  Complete Development to Modify Application | The Team will complete the development of the application and test the error trapping and user functionality. |
| Week 17  AM session | Sprint Two | Question Fourteen.  **Finalise Sprint One** | The second Scrum Master will present the application with Sprint Two Program Criteria. The lecturer/assessor can ask questions. |
| PM session | Sprint Two | **Assessment Point for second 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 18  AM session | Handover |  |  |
| PM Session | Handover |  |  |
| Week 19  AM session | Handover |  |  |
| PM Session | Handover |  |  |

## 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.

## 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

* A button to calculate the mid-extreme,
* A button to calculate the mode,
* A button to calculate the average,
* A button to calculate the range.
* All GUI components have detailed tool tips.
* A button to initiate a sequential search.

### Program Functionality

* The mathematic calculations will display in separate text boxes formatted to 2 decimal places as appropriate.
* The mathematical calculations must be coded using custom algorithms.
* The sequential search (was sort) method must be coded using a single FOR loop and one IF condition.
* 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.

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

## 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 | Visual Studio Community 2022 |
| Application Development IDE | Visual Studio Community 2022 |
| UI Components | |
| |  |  | | --- | --- | | Component | Event | | Text Box (Input) | Non read only Text Box for user to input the integer to search for and to display the chosen integer from the List Box if needed to edit. | | Text Box (Message) | Read Only Text Box to display normal and error messages to the user when needed. | | List Box | Will Display all items from the Array for user to view. | | Button (Search) | When clicked a Binary Search Algorithm will be used to search Array for integer present in Text Box. | | Button (Sort) | When clicked a Bubble Sort Algorithm will sort the array from minimum value to maximum value. | | Button (Edit) | When clicked if a chose value from the Array has been changed the Array will update with the edited value replacing the previous value. | | Label (Input Box) | Labels editable Text Box as Input Box. | | Label (List Box) | Labels the List Box as Neutrino Interactions. | | Label (Message Box) | Labels the non-editable Text Box as Message Box. | | Button (Sequential Search) | When clicked will parse the input from user to integer. Will then iterate through the array and compare each value to the key (parsed value). If a match is found return true, else return false. Message box will be displayed to confirm search result. | | Button (Average) | Will find the sum of all array values and divide it by the total amount of numbers added. Result displayed in textBoxResult. | | Button (Mode) | Will find the number(s) that occur the most. Will have an inner and outer for loop. Will cycle through each number and compare it to all the other numbers in the array and add to a counter for each matching number. Number that occurs the most will be added to a list<>. If another number occurs the same amount, it will also appear in the list. Result(s) will be displayed in textBoxResult | | Button (Range) | Will find the highest and lowest number in the array and find the difference between them. Result will be displayed in textBoxResult | | Button (Mid-Extreme) | Will find the highest and lowest numbers in the array and find the mid point between them. Result will be displayed in textBoxResult. | | Textbox (textBoxResult) | Read-only text box to display the results of mathematical operations. | | Label (Data Processing Result) | Lables textBoxResult so user knows where to expect mathematical operations outputs. | | |
| GUI Layout | |
|  | |

## 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:

* 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.
* Mode: The mode is defined as the number that appears most frequently in a set of data (unimodal).
* Average: The average is defined as the sum of all the values divided by the total number of values in the data set.
* 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 | |
| bool SequentialSearch(int[] array, int key)  {  foreach (int i in array)  {  if (array[i] == key)  {  return true;  }  }  return false;  } | FOR EACH integer **i** in **array**  IF **I** equals **key**  RETURN true  END IF  ENDFOR  RETURN false |
| Algorithm/Pseudo Code Design for Mathematical Functions | |
| Mid-Extreme:  int max = array[0];  int min = array[0];  double mid;  foreach(int i in array)  {  if (i >= max)  {  max = i;  }  else if(i <= min)  {  min = i;  }  }  mid = (max - min) / 2;  return mid;  Mode:  List<int> Mode(int[] array)  {  int maxCount = 0;  List<int> modes = new List<int>();  foreach(int i in array)  {  int count = 0;  foreach(int j in array)  {  if ((i == j))  {  count++;  }  }  if (count > maxCount)  {  maxCount = count;  modes.Clear();  modes.Add(i);  }  else if (count == maxCount)  {  modes.Add(i);  }  }  return modes;  }  Average:  double Average(int[] array)  {  int sum = 0;  int size = 0;  foreach (int i in array)  {  sum += array[i];  size++;  }  return sum/size;  }  Range:  double Range(int[] array)  {  int min = array[0], max = array[0];  double range;  foreach (int i in array)  {  if (i <= min)  {  min = i;  }  if ((i >= max))  {  max = i;  }  }  range = max - min;  return range;  } | Mid-Extreme  INITIALISE integer **max** to array index 0  INITIALISE integer **min** to array index 0  INITIALISE double **mid**  FOR EACH integer **i** in **array**  IF **i** is equal or less than **min**  **min** = **i**  ENDIF  ELSEIF **i** is equal or more than **max**  **max** = **i**  ENDELSEIF  ENDFOREACH  **mid** = (**max**-**min**)/2  RETURN **mid**  Mode:  INITIALISE "**maxCount**" to 0  INITIALISE new integer list "**modes**"  FOREACH integer "**i**" in "**array**"  SET "**count**" to 0  FOREACH integer "**j**" in "**array**"  IF "**i**" equals "**j**" increment "**count**" up by 1  ENDIF  ENDFOREACH  IF "**count**" is greater than "**maxCount**"  SET "**maxCount**" to "**count**" value  CLEAR list "**modes**"  ADD "**i**" to "**modes**"  ENDIF  ELSEIF "**count**" is equal to "**maxCount**"  ADD "**i**" to "**modes**"  ENDELSEIF  ENDFOREACH  RETURN "**modes**"  Average:  INITIALISE "**sum**" to 0  INITIALISE "**size**" to 0  FOREACH integer "**i**" in "**array**"  INCREMENT "**sum**" by "**array**" index "**i**" value  INCREMENT "**size**" by 1  ENDFOREACH  RETURN "**sum**" divided by "**size**"  Range:  INITIALISE integer **max** to array index 0  INITIALISE integer **min** to array index 0  INITIALISE double **range**  FOR EACH integer **i** in **array**  IF **i** is equal or less than **min**  **min** = **i**  ENDIF  ELSEIF **i** is equal or more than **max**  **max** = **i**  ENDELSEIF  ENDFOREACH  Integer **range** = **max** – **min**  RETURN **range** |

## 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; |  |

## 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 for approval 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.

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

* Mean: The mean is defined as the sum of the smallest value and the largest value in the given data set divided by 2.
* Mode: The mode is defined as the number that appears most frequently in a set of data (unimodal).
* Average: The average is defined as the sum of all the values divided by the total number of values in the data set.
* Range: The range is defined as the difference between the largest and the smallest values in the data set.

## 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:

* The program must load random data into the array using a button click.
* The client must be able to enter search data and click a button to activate a linear search.
* The client can click each function button which will populate the related textbox.
* The mathematical results are properly formatted.
* All major components have detailed tool tips.
* All error messages are demonstrated.
* Code comments are relevant.
* Answer questions on all aspects of the Sprint Two development and documents.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sprint Two  Assessment Task Three  Marking Guide and Observation Checklist | | | | |
| Student Name  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Tick one | | | |
| Scrum Master | | | Team Member |
| Demonstration Criteria | Satisfactory  Yes No | | Comment | |
| 1. The program must load random data into the array using a button click. |  |  |  | |
| 1. The client must be able to enter search data and click a button to activate a linear search. |  |  |  | |
| 1. The client can click each function button which will populate the related textbox. |  |  |  | |
| 1. The mathematical results are properly formatted. |  |  |  | |
| 1. All major components have detailed tool tips. |  |  |  | |
| 1. All user input messages are demonstrated. i.e. Incorrect integer input (ie “A”, char) |  |  |  | |
| 1. Code comments are relevant. |  |  |  | |
| 1. Three Completed Scrum Boards    1. history of each Project Board    2. designed has been checked to see that the code has been developed, designed, reviewed and maintained over time. |  |  |  | |
| 1. Project Specifications completed as per the organisational template provided. |  |  |  | |
| Observation Checklist | Satisfactory  Yes No | | Comment | |
| 1. Participated fully in team discussion/meeting in all aspects of the development process |  |  |  | |
| 1. Followed the appropriate software development standards and workflow when creating all code (using an IDE to write, document, test, and debug) |  |  |  | |
| 1. Create and share GitHub resources - assessor to ensure GitHub versioning and history is checked. |  |  |  | |
| 1. Asked appropriate questions in relation to all areas of design and development |  |  |  | |
| 1. Listens and responds accordingly to team members |  |  |  | |
| 1. Speaks clearly using technical language ensuring that the project progresses. |  |  |  | |
| 1. Discusses critically aspects of technical specifications |  |  |  | |
| 1. Collaborated in the Project Specifications document |  |  |  | |
| 1. Collaborated with coding and app development as confirmed on GitHub |  |  |  | |
| 1. Uses and develops standard algorithms and MSDN language standards |  |  |  | |
| 1. Applies modular programming principles separating the functionality of a program into independent, interchangeable modules |  |  |  | |
| 1. Completes designated tasks from Project Board as confirmed on GitHub |  |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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