

# Assessment Task

## - Written



<b>Student Name</b>		<b>Student Number</b>	
<b>Unit Code/s &amp; Name/s</b>	ICTPRG418 Apply intermediate programming skills in another language		
<b>Assessment Type</b>	Written		
<b>Assessment Name</b>	<b>Written Assignment</b> Programming Assignment 2	<b>Assessment Task No.</b>	AT2
<b>Assessment Due Date</b>	6/09/18	<b>Date submitted</b>	
<b>Assessor Name</b>			
<p><b>Student Declaration:</b> I declare that this assessment is my own work. Any ideas and comments made by other people have been acknowledged as references. I understand that if this statement is found to be false, it will be regarded as misconduct and will be subject to disciplinary action as outlined in the TAFE Queensland Student Rules. I understand that by emailing or submitting this assessment electronically, I agree to this Declaration in lieu of a written signature.</p>			
<b>Student Signature</b>		<b>Date</b>	
<p><b>PRIVACY DISCLAIMER:</b> TAFE Queensland is collecting your personal information for assessment purposes. The information will only be accessed by authorised employees of TAFE Queensland. Some of this information may be given to the Australian Skills Quality Authority (ASQA) or its successor and/or TAFE Queensland for audit and/or reporting purposes. Your information will not be given to any other person or agency unless you have given us written permission or we are required by law.</p>			

## Instructions to Student

### Learning Support

Additional support is available to help you achieve your learning goals. Speak to your teacher or a Learning Skills Centre team member if you feel that you may benefit from some extra support. The Institute provides extra support through the Disability Support Unit and the Learning Skills Centre.

RPL (Recognition of Prior Learning) is available for this unit. Speak to your teacher/assessor to check if you qualify for RPL.

### Conditions of Assessment

You will need to complete the learning and undertake all assessments satisfactorily to be deemed competent. You are responsible for complying with all assessment item instructions; submission and collection requirements; undertaking assessment tasks honestly and retaining a copy of all assessment items.

You must submit assessment items by the **due date**, unless an extension has been granted by your teacher. Failure to submit assessment items by the due date will result in a "did not submit" being recorded and depending on your circumstances, you may be granted one final resubmission.

To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.

### The Classroom as a Simulated Work Environment

Students must be aware and take responsibility for the problems of working in a shared IT environment. Problems such as noise levels, production flow, interruptions and time variances are common to workplaces. In the simulated environment provided in the classroom these problems can take the form of:

- Other students who continually ask questions or talk aloud while thinking
- Fire drills, projector not working, printers running out of paper or toner cartridge
- Miscalculating how much work you can do in one day, missing classes and so on.

Some things are unavoidable and you must devise strategies to overcome them, for example, we cannot stop students from asking questions or entering at exiting the class. Other things are unpredictable (e.g. fire drills). You need to be aware and plan and organise your work allowing some extra time for unavoidable and unpredicted events.

	<p><b>Assessment Criteria:</b></p> <p>To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate key skills/tasks/knowledge to an acceptable industry standard.</p> <p>Refer to the marking criteria document for a detailed list of items.</p> <p><b>Number of Attempts:</b></p> <p>You will receive up to two (2) attempts at this assessment task. Should your 1<sup>st</sup> attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant sections / questions with you and will arrange a due date for the submission of your 2<sup>nd</sup> attempt. If your 2<sup>nd</sup> submission is unsatisfactory (U), or you fail to submit a 2<sup>nd</sup> attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task, with the exception of Apprentices or Trainees who are permitted an additional supplementary assessment. <b>For more information, refer to the Student Rules.</b></p>
<b>Submission details</b>	<p>Submit your assessment to the allocated dropbox in <b>Connect</b> or to the allocated network folder.</p> <p>Your teacher will provide all the details for the submission system or network.</p> <p>Your assignment must be saved with your surname_student number_unit/cluster_AssessmentNumber. For example:</p> <p><b>surname_1234567890_ICTPRG520_1</b></p> <p>For re-submissions, an “R” must be added to the file name. For example:</p> <p><b>surname_1234567890_ICTPRG520_1_R</b></p> <p>The Marking Criteria Sheet must be signed and submitted with your work.</p>
<b>Instructions for the Assessor</b>	<p>To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.</p> <p>The assessment must be conducted in safe conditions replicating the workplace. Noise levels, production flow, interruptions and time variances must be typical of those experienced in the programming and software development field of work and include access to:</p> <ul style="list-style-type: none"> <li>• a software development environment</li> <li>• the technical requirements as outlined in the task instructions</li> </ul>
<b>Note to Student</b>	<p>An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide.</p>

# Instructions to Students

## Assignment Scenario: Caroline's Classroom Robots

### Original Client Requirement *(This detail is provided for background information purposes only):*

Caroline is looking to get your support in programming one or more robots to assist her with classroom administration and supporting student activity.

Specifically, Caroline would like the robot(s) to assist her with:

- Welcoming students to the class and tracking their attendance.
- Interacting with students in the delivery of a range of tutorial exercises.
- Tracking the activities and results of students throughout each lesson.
- Store a record of these various details for review, reporting and statistical reporting purposes; and for planning future class exercises, activities and preparation.

Caroline would like screens and reports accessible via her computer tablet to review these various details on a day-to-day basis, and for reviewing periodic reports.

### Current Client Requirement:

After further consideration, Caroline has a further request - discussed below and on the following pages.

### Programming Activity:

Caroline has asked whether it may be possible to have an application that displays the layout of her classroom, and provide her with the ability to:

1. **View** the layout of the room including desks and student names in a graphical (grid-style) format.
2. **Edit** these details on screen.
3. **Clear** all the student names.
4. **Save** the updated details into a new file. This will require the application to also save an updated teacher name, class, room and date.
5. Select a required data **file** via an Open-File dialog box.
6. **Sort** the student list alphabetically by name, and present the resulting list on a popup dialog box with each student's grid location noted.
7. **Search** for a student and have their location highlight within the grid. Then display the sorted list of students *(as per dot point 6 above)* and highlight the required person. *(You need to search for the required student within the sorted list using a **binary search**.)* *(See screen image on the following page.)*
8. Save the data into a **Random Access File**, and implement a method for reading back a specific random access file entry.

Classroom Plan

Teacher: **Caroline**

Class: **5B**

Room: **B16**

Date: **7/03/18**

	0	1	2	3	4	5	6	7	8	9
0					Front Desk					
1										
2										
3										
4										
5		Jane	Alisha		Tony	Ahmed		Peta	Jenny	
6										
7										
8		Hanna	Terri		James	Henri		Cian	Ana	
9										
10										
11		Han	Mark		Sian	Sarah		Makin	John	
12										
13										
14										
15										
16										
17										
18										

Clear

Save

Sort

Search

Find

BAF

Exit

TAFE Queensland 2018

## Popup Sort/Search Screen

Student List - Search: Henri		
Student	Across	Down
Ahmed	5	5
Alisha	2	5
Ana	8	8
Cian	7	8
Han	1	11
Hanna	1	8
Henri	5	8
James	4	8
Jane	1	5
Jenny	8	5
John	8	11
Makin	7	11
Mark	2	11
Peta	7	5
Sarah	5	11
Sian	4	11
Terri	2	8
Tony	4	5

TAFE Queensland 2018

## Sample Data:

A sample data file has been provided within the assignment folder. *(Rename this file as required.)*

Filename: **2\_TheClassroomLayout\_SampleData.csv**

A portion of this data file is as follows:

Teacher	Caroline		
Class	5B		
Room	B16		
Date	7/03/2018		
4	0	Front Desk	
3	1	BKGRND FILL	blue
4	1	BKGRND FILL	blue
5	1	BKGRND FILL	blue
6	1	BKGRND FILL	blue
1	4	BKGRND FILL	blue
2	4	BKGRND FILL	blue
4	4	BKGRND FILL	blue
5	4	BKGRND FILL	blue
7	4	BKGRND FILL	blue
8	4	BKGRND FILL	blue
1	5	Jane	
2	5	Alisha	
4	5	Tony	
5	5	Ahmed	
7	5	Peta	
8	5	Jenny	
1	7	BKGRND FILL	blue
2	7	BKGRND FILL	blue
4	7	BKGRND FILL	blue

**Your tasks: (NOTE: Items 1.1 - 1.8 are repeated from page 2 above)**

Caroline requires this application to display the layout of her classroom, and provide her with the ability to:

- |     |                                                                                                                                                                                                                                                                                                                                                           |  |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 1.1 | <b>View</b> the layout of the room including desks and student names in a graphical (grid-style) format.                                                                                                                                                                                                                                                  |  |
| 1.2 | <b>Edit</b> these details on screen.                                                                                                                                                                                                                                                                                                                      |  |
| 1.3 | <b>Clear</b> all the student names.                                                                                                                                                                                                                                                                                                                       |  |
| 1.4 | <b>Save</b> the updated details into a new file. This will require the application to also save an updated teacher name, class, room and date.                                                                                                                                                                                                            |  |
| 1.5 | Select a required data <b>file</b> via an Open-File dialog box.                                                                                                                                                                                                                                                                                           |  |
| 1.6 | <b>Sort</b> the student list alphabetically by name into a secondary data structure, and present the resulting list on a popup dialog box with each student's grid location noted. Add code that will maintain this sorted list as student details are adjusted on screen.                                                                                |  |
| 1.7 | <b>Search</b> for a student and have their location highlight within the grid. Then display the sorted list of students ( <i>as per item 1.6 above</i> ) and highlight the required person. ( <i>You need to search for the required student within the sorted list using a <b>binary search</b>.</i> ) ( <i>See screen image on the previous page.</i> ) |  |
| 1.8 | Save the data into a <b>Random Access File</b> ( <i>and implement a method for reading back a specific random access file entry</i> ).                                                                                                                                                                                                                    |  |

Additionally, you are required to:

- |      |                                                                                                                                                                                                                                                                                                                                                                                                                                               |  |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 1.9  | Have appropriate <b>design and technical documentation</b> , a <b>test plan</b> , <b>test cases plus the results</b> of testing. The <b>source code</b> is to be supplied and should contain appropriate in-line comments. Your programming should follow your development team's <b>programming standards</b> . In line with this, selected components are to be located in and accessed from <b>secondary program (source-code) files</b> . |  |
| 1.10 | <b>Prepare and review your internal and external documentation</b> for your application, including a user manual. Ensure that it aligns with organisational documentation standards provided, contains correct grammar and spelling, and is appropriate for developers who may provide support to this application in the future.                                                                                                             |  |

- 1.11 Prepare **automated program documentation** using a facility provided within your IDE, or using a separate facility such as Javadoc.
- 1.12 Provide evidence of your use of the **debugging facilities** with the Integrated Development Environment (IDE) you are utilising. This may be demonstrated with a series of screen images of you debugging your application within your selected IDE – showing at least one breakpoint, a set of associated watches, and you tracing through several lines of code.
