Assessment Marking Criteria

Student Name	Kyle Kent	**Student Number**	465510139
Unit Code/s & Name/s	ICTPRG418 Apply intermediate programming skills in another language		
Assessment Type	Written		
Assessment Name	**Written Assignment** Programming Assignment 2	**Assessment Task No.**	AT2
Assessment Due Date	*6/09/2018*	**Date submitted**	*6/09/2018*

Assessor Feedback:

☐ **Student provided with feedback** *(check box when completed)*

Attempt 1	Satisfactory ☐	Unsatisfactory ☐	**Date**	/ /
Attempt 2	Satisfactory ☐	Unsatisfactory ☐	**Date**	/ /

Assessor Name		**Assessor Signature**	

Note to assessor: Please record below any reasonable adjustment that has occurred during this assessment e.g. written assessment given orally.

Note: The Assessment Conditions for this unit are:

Gather evidence to demonstrate consistent performance in conditions that are safe and replicate 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:

- Software development environment
- Technical requirements.

Assessors must satisfy NVR/AQTF assessor requirements.

Assessment criteria / benchmarks **The evidence submitted demonstrates that the student satisfactorily:**	**Attempt 1**		**Attempt 2**	
	S	**U**	**S**	**U**
1.1 - **View** the layout of the room including desks and student names in a graphical (grid-style) format.	☐	☐	☐	☐
1.2 - **Edit** these details on screen.	☐	☐	☐	☐
1.3 - **Clear** all the student names.	☐	☐	☐	☐
1.4 - **Save** the updated details into a new file. This will require the application to also save an updated teacher name, class, room and date.	☐	☐	☐	☐
1.5 - Select a required data **file** via an Open-File dialog box.	☐	☐	☐	☐
1.6 - **Sort** the student list alphabetically by name, and present the resulting list on a popup dialog box with each student's grid location noted.	☐	☐	☐	☐
1.7 - **Search** for a student and have their location highlight within the grid. Then display the sorted list of students *(as per item 1.6 above)* and highlight the required person. *(You need to search for the required student within the sorted list using a **binary search**.)*	☐	☐	☐	☐
1.8 - Save the data into a **Random Access File** *(and implement a method for reading back a specific random access file entry)*.	☐	☐	☐	☐
1.9 - Have appropriate design and technical documentation, a test plan, test cases plus the results of testing. The source code is to be supplied and should contain appropriate in-line comments. Your programming should follow your development team's programming standards. In line with this, selected components are to be located in and accessed from secondary program (source-code) files.	☐	☐	☐	☐
1.10 - Review your **internal and external documentation** for your application. 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.	☐	☐	☐	☐
Evidence of the student having demonstrating consistent performance:				
• In developing in a software development environment	☐	☐	☐	☐
• Using technical requirements	☐	☐	☐	☐
The student has demonstrated competency in:				
• Task skills - performing every task in the assessment at an appropriate skill level	☐	☐	☐	☐
• Task management skills - managing the various tasks in this assessment at an appropriate skill level, and within a working environment that replicates a workplace appropriate for software development.	☐	☐	☐	☐
• Contingency management skills - managing issues that arise at an appropriate skill level, and within a working environment that replicates a workplace appropriate for software development.	☐	☐	☐	☐
• Job role environment skills – operating and interacting appropriately and effectively within a working environment that replicates a workplace appropriate for software development.	☐	☐	☐	☐
Note: These checklist items are based on additional Critical Evidence, Required Skill and Required Knowledge not explicitly stated elsewhere in the assignment requirements checklists.				