Assessment Task
- Written

Student Name		Student Number	
Unit Code/s & Name/s	ICTPRG418 Apply intermediate programming skills in another language		
Assessment Type	Written		
Assessment Name	**Written Assignment** Programming Assignment 1	Assessment Task No.	AT1
Assessment Due Date	17/8/18	Date submitted	
Assessor Name			
Student Declaration: 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.			
Student Signature		Date	
PRIVACY DISCLAIMER: 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.			

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.

	Assessment Criteria:
	To achieve a satisfactory result, your assessor will be looking for your ability to demonstrate key skills/tasks/knowledge to an acceptable industry standard.
	Refer to the marking criteria document for a detailed list of items.
	Number of Attempts:
	You will receive up to two (2) attempts at this assessment task. Should your 1st 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 2nd attempt. If your 2nd submission is unsatisfactory (U), or you fail to submit a 2nd 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. **For more information, refer to the Student Rules.**
Submission details	Submit your assessment to the allocated dropbox in **Connect** or to the allocated network folder.
	Your teacher will provide all the details for the submission system or network.
	Your assignment must be saved with your surname_student number_unit/cluster_AssessmentNumber. For example:
	surname_1234567890_ICTPRG418_1
	For re-submissions, an "R" must be added to the file name. For example:
	surname_1234567890_ ICTPRG418_1_R
	The Marking Criteria Sheet must be signed and submitted with your work.
Instructions for the Assessor	To be judged competent in this assessment item the student is required to demonstrate competence in all indicators shown in the marking guide.
	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:
	• a software development environment
	• the technical requirements as outlined in the task instructions
Note to Student	An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide.

Instructions to Students

'My-Friend Tracker' Software Development Assignment

Case Study Script and Requirements

Sasha has asked you to prepare a small C# application that might be used for **tracking friends, clients and family names, likes, dislikes and birth-dates**.

The aim is to provide Sasha with a report per month of upcoming birthdays including reminders as to individual's likes and dislikes - assisting with the selection of presents.

The application will allow Sasha to:

- Enter new person records - names, likes, dislikes, day-of-birth and month-of-birth.

- Use various navigation buttons:

  - First, last, next and previous record buttons

  - A Find button to search for an existing record by person name

- Use New, Save and Delete buttons.

- View upcoming birthdays for a selected month.

- Use a second search method for finding an existing record by person name, within the sorted list, using a binary search.

The application will save and load the people data to and from a comma-delimited csv file: **MyFriendData.csv**.

Java screenshot used with permission from Oracle.

Your tasks:

Sasha requires this application to:

1.1   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.

1.2   **Load** all the current person records from the MyFriendData.csv data file on the hard drive (or equivalent).

1.3   **Add** new person records and **Edit** and **View** existing person records - names, likes, dislikes, day-of-birth and month-of-birth.

1.4   Provide **navigation buttons** and a **search** facility to navigate through the various person records.

1.5   Allow the program user to **save** all the person records back to the hard drive (or equivalent). Save all the record once the program is closed.

1.6   Display a **list** of the upcoming birthdays for the selected month.

1.7   Allow the user to enter a person name and then:

- Present a **sorted** list of all people, and…

- Quickly **search** for and present the order that the selected person is in the sorted list.

1.8   Be provided with a supporting **user manual** appropriately considered and structured with correct spelling and grammar.

Additionally, you are required to:

1.9   **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.

1.10   Prepare **automated program documentation** using a facility provided within your IDE, or using a separate facility such as Javadoc.

1.11   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.