

Assessment Task

– Portfolio of Evidence



Student Name	Kyle Kent	Student Number	465510139
Unit Code/s & Name/s	ICTPRG527 Apply intermediate object-oriented language skills		
Assessment Name	Written Assignment Testing Assignment	Assessment Task No.	AT1
Assessment Due Date		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	K Kent	Date	20/09/2018

Instructions to Student	<p>Learning Support</p> <p>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.</p> <p>RPL (Recognition of Prior Learning) is available for this unit. Speak to your teacher/assessor to check if you qualify for RPL.</p> <p>Conditions of Assessment</p> <p>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.</p> <p>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.</p>
--------------------------------	---

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 (if relevant)	<p>Submit your assessment to the allocated dropbox in Connect 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>surname_1234567890_ICTPRG527_1</p> <p>For re-submissions, an “R” must be added to the file name. For example:</p> <p>surname_1234567890_ICTPRG527_1_R</p> <p>The Marking Criteria Sheet must be signed and submitted with your work.</p>
Instructions to Assessor	<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>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 industry, and include access to:</p> <ul style="list-style-type: none"> • The programming specifications • Programming languages for object-oriented applications • Database management system (DBMS) • Resources to assist in doing the assessments and learning <p>Note: A project specification document needs to be presented to or needs to be made available for the students to proceed with their testing.</p>
Note to Student	<p>An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide.</p>

Assessment 1: Software Programming

In this assessment, you are required to participate in the development of an object-oriented application using an object-oriented programming language of your choice or chosen by your Project Manager.

You are required to show clear evidence of due process – appropriate planning, gathering requirements, design, development, debugging, testing, and documentation of the object-oriented application, reporting to your project manager on updates required and completion of the project.

Your tasks:

1. Create and send an email to your Project Manager to obtain program specifications, business case scenario and programming guidelines. Gather other requirements and review the documents in preparation for the development of the project.
2. **Based** on the scenario above and your **interpretation** of the **program specifications** and **gathered requirements**:
 - a. **Create two projects** in your **Project Solution** that will do the following:
 - i. **One project** as a Class Library to **package** and **implement** a **class** that will **handle** and manage **database connectivity** and **transactions**
 - ii. **One project** as the Application project to **package** and **implement** a **class** that will **handle** and manage the **application's Graphical User Interface (GUI)** framework.
 - b. Build the two projects to **create** a **binary file objects** in the form of .DLL and an .EXE and ensure that the .EXE file can subscribe to or consume the .DLL files.
 - c. **Make use** of the **app.config** file to **store internal data** such as the database connection strings, image location, and other data that predefined the configurations of your application.
 - d. **Make use** of **Application Settings (Properties->Settings)** to **store internal data** about user preferences or any other information your application needs such as background colour of forms or preferred fonts.
 - e. **Create macros** in your chosen **integrated development environment (IDE)** to **automate repetitive programming actions** and **program building**.
3. **Based** on the scenario above and your **interpretation** of the **program specifications** and **gathered requirements**, **develop** the **Class Library** (database connectivity) project that will do the following:
 - a. **Use** and **search** the **app.config** file for the connection string to **connect** the **application to** a **database**.
 - b. Make **use** of **error trapping techniques** to trap logical errors and action as necessary.
 - c. Make **use** of **methods** and **method overloading** to either **retrieve, insert, update, or delete data stored** in the **database** ensuring to **maintain transactional integrity** by **rolling back database updates when error occurs**.
 - d. Make **use** of **methods** to **manipulate database schema** such as CREATE TABLE, ALTER TABLE or DROP TABLE.

- e. **Write code** that **follows** the **coding standards** and **naming conventions outlined** in the **programming guidelines** you have obtained in Task 1 and ensure to **include comment-blocks** for user-defined methods.
4. **Based** on the scenario above and your **interpretation** of the **program specifications** and **gathered requirements**, **develop** the **application** project **using** the **Graphical User Interface (GUI) framework** that will do the following:
 - a. **Sort and search Application Settings (Properties->Settings)** to implement user preferences or program settings upon start-up of the application.
 - b. **Make use of GUI components:** forms, textboxes, labels, buttons, combo boxes, data grids, and other components that your application needs.
 - c. **Ensure** that the **GUI components respond** to **user's actions** such as on button clicked, on double clicked of the data grid, on text changed and other GUI events.
 - d. **Make use of error trapping techniques** to trap logical errors and action as necessary.
 - e. **Make use of a class** that **inherits from either multiple classes** or **class Interface** by either incorporating it to this project or developing a stand-alone application to demonstrate multiple inheritance.
 - f. **Write code** that **follows** the **coding standards** and **naming conventions outlined** in the **programming guidelines** you have obtained in Task 1 and ensure to **include comment-blocks** for user-defined methods.
5. To **ensure** all **syntax** and **logic errors** are **identified** and the debugging process, outcomes and corrections comply with program specification:
 - a. **Use the debugging tools** including trace and watches to create a log of syntax and logic errors
 - b. **Outline the solutions applied**
 - c. Create a technical report and include this under the section "Defect Logs".
 - d. Include screenshots and steps of using these tools under the section "Debugging Tools in Action".
6. **Create** and **prepare test data** to confirm code meets design specifications.
7. **Perform testing using the prepared test data** and **document the test results** and compare against program specifications and gathered project requirements. Have at least one of your peers to test the program and
 - a. **Document the test results**
 - b. **Analyse results**, and...
 - c. **Prepare a test summary report.**

8. Include a section in your technical report titled: **“Documentation Maintenance”** to indicate how you will **maintain** the **program documentation**. Include in this section your approaches to managing document version control, and naming conventions
9. Present your development project, including technical report, to your Project Manager for sign-off.