

# 2022 Subject & Assessment Guide

## Code Design and Data Structures

ICT50220 Diploma of Information Technology  
(Game Programming)

CUA51020 Diploma of Screen and Media

# Table of Contents

Code Design and Data Structures .....	3
Units of Competency .....	3
Overall Learning Outcomes.....	3
Subject Description .....	3
Industry Relevance.....	4
Assumed Knowledge.....	4
Subject Textbooks .....	4
Assessment Criteria .....	5
Assessment Description .....	5
Software.....	8
Core.....	8
Suggested.....	9

# Code Design and Data Structures

## Units of Competency

The units of competency that are covered in this subject are as follows:

[ICTPRG547](#) - Apply advanced programming skills in another language

[CUADIG512](#) - Design digital applications

Assessment processes and competency evidence requirements are described in the *Assessment Criteria* section below. If you have prior or other evidence against competency you should discuss this with your teacher.

## Subject Overview

### Overall Learning Outcomes

- Apply a knowledge of object-oriented design and modelling
- Demonstrate an understanding of common data structures used in game programming and their application
- Demonstrate an understanding of fundamental programming algorithms and their application
- Apply skills in testing and error handling in game programming

### Subject Description

This subject is designed to teach you about the design techniques that underpin modern programming, particularly object-oriented programming (OOP), and techniques commonly used in games programming today.

OOP grew to maturity in the early 1990's and quickly revolutionised how programs were designed and developed. Before that time, the prevailing software design technique was *functional decomposition* (also known as procedural programming) which was all about writing *functions* that implemented the *behaviour* of a program. It was soon realised that this technique was unsuitable for very large software systems and that a better approach was needed.

The new approach became known as *object-orientation*, a term used to describe both the new design methodology and the coding techniques used to implement object-oriented designs in an actual programming language. C++ was developed as an extension of the earlier C language with the express purpose of implementing the new object-oriented principles and techniques.

This subject looks at the design and implementation of common *algorithms* (such as sorting and searching), *data structures* (used to store data efficiently for processing), and systems (including Game State management and Component patterns) used in many games and game engines.

## Industry Relevance

Object-oriented design is now the primary design methodology used by almost all modern programming languages in commerce and industry. OOP is vital in the quest to write ever larger and more complex software systems. Skills in OOP design are highly valued and sought after throughout the software industry, including the games industry. Employers are no longer looking only for coders; they are looking for designers / software engineers, i.e. programmers who know how to develop a design and then implement it in effective and maintainable code.

This importance extends even further to the lower-level data structures examined in this course. Once you are actually implementing a software design, it is vitally important to then know which algorithms and data structures to use and when to use them. This translates to understanding of how algorithms and data structures work and their relative efficiencies. A brilliant design is only as good as its actual implementation.

## Assumed Knowledge

- An understanding of foundational programming concepts and knowledge of C++ at an introductory level

## Subject Textbooks

The following textbooks are highly recommended for this subject:

- Prata, S, **C++ Primer Plus**, 6th Edition, Addison Wesley (2011)
- Nystrom, R, **Game Programming Patterns**, 1st Edition, Genever Benning (2014)
  - <http://gameprogrammingpatterns.com/>

You may also find the following textbooks useful:

- Sherrod, A, **Data Structures and Algorithms for Game Developers**, 1st Edition, Charles River Media (2007)
- Dalmau, D, **Core Techniques and Algorithms in Game Programming**, 1st Edition, New Riders Games (2003)
- Sedgewick, R, Wayne, K, **Algorithms, 4th Edition**, Addison-Wesley Professional (2011), <https://algs4.cs.princeton.edu/home/>

# Assessment Criteria

## Assessment Description

### Assessment Milestones

***Please refer to your Class Schedule for actual dates on your campus***

### General Description

This assessment requires you to demonstrate your knowledge of various complex data structures and algorithms used within games and simulation industries by implementing and testing each technique.

You are required to complete several assessable tasks throughout this subject.

- Design game optimisations
- Implement a double-linked list
- Implement a binary tree
- Use hashing techniques
- Demonstrate debugging techniques
- Complete the Interprocess Communication exercise
- Design a classic game

On *Canvas* you will find briefs for each of these tasks. Some tasks may have multiple briefs, allowing you to select the activity to complete to demonstrate your competency in the requisite task. Similarly, some tasks may be combined into a single project. All programming tasks must be completed as C++ programs.

### Evidence Specifications

This is the specific evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed in the table below. You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your teacher.

### Your Roles and Responsibilities as a Candidate

- Understand and feel comfortable with the assessment process.
- Know what evidence you must provide to demonstrate competency.
- Take an active part in the assessment process.
- Collect all competency evidence for presentation when required.

This table defines the individual requirements for each part of the assessment criteria. Listed here are the cumulative requirements for all assessment items. The evidence requirements for specific assessment items can be seen by referring to the table listed for that assessment item in the following sections.

## Assessment and Competency Requirements

### 1. Design Game Optimisations

Evidence that includes:

- A 2-page brief detailing data structure and algorithm updates to improve the performance of the provided program. Your brief is to include:
    - A basic class diagram
    - A description of the data structures to be implemented
    - A description of the algorithms to be implemented
    - A description of testing strategies that may be employed to verify the implementation of data structures or algorithms or measure performance increases
    - A description of any other programming or feature changes to be implemented
- Implementation of the designed changes is not required. However, any implemented changes may be used as evidence of competency for other assessment tasks.

### 2. Implement a Double-Linked List

Evidence that includes:

- Successful creation of an application demonstrating the implementation of a double-linked list
- Project submitted as an executable binary file that can be run external to an IDE
- Source code and project files submitted for review

### 3. Implement a Binary Tree

Evidence that includes:

- Successful creation of an application demonstrating the implementation of a binary tree
- Project submitted as an executable binary file that can be run external to an IDE
- Source code and project files submitted for review

### 4. Use Hashing Techniques

Evidence that includes:

- Successful creation of a project which implements and demonstrates hashing techniques
- Creation of a custom hashing function
- Project submitted as an executable binary file that can be run external to an IDE
- Source code and project files submitted for review

### 5. Perform Unit Testing and Debugging

Evidence that includes:

- Design and creation of tests using a testing framework
- Use of tests to debug and verify the implementation of one or more data structures
- Test report documenting the tests conducted and the test results

### 6. Inter-Process Communication Exercise

Evidence that includes:

- Completion of the inter-process communication exercise
- Project submitted as an executable binary file that can be run external to an IDE
- Source code and project files submitted for review

### 7. Design a Classic Game

Evidence that includes:

- A 3- to 5-page brief detailing the design of a classic 2-D arcade game. Your brief is to include:
  - A brief description of the game
  - A basic class diagram
  - A description of any programming patterns used in the design
  - A description of the data structures to be implemented
  - A description of the algorithms to be implemented
  - A description of applicable testing strategies
  - Mock-ups of the user interface and menus
- Implementation of the game is not required. Your brief must be detailed enough for another programmer to read and follow

### 8. Application Handover

Evidence that includes:

- Visual Studio solutions and projects that compile without errors
  - All temporary and built executable files in the obj and bin folder are removed
- A “readme” or client document explaining how to compile, run and operate the program for each application made
- All submitted material archived in a single compressed file (zip, rar, or 7z)

## Assessment Instructions for Candidate

### METHOD OF ASSESSMENT

Assessment is a cumulative process which takes place throughout a subject. A ‘competent’ or ‘not yet competent’ decision is generally made at the end of a subject. Your assessment will be conducted by an official AIE qualified assessor. This may be someone other than your teacher. The evidence you must prepare and present is described.

above in this assessment criteria document. This evidence has been mapped to the units of competency listed at the beginning of this document. Assessments will be conducted on a specific milestone recorded above in this assessment guide document.

### ASSESSMENT CONDITIONS

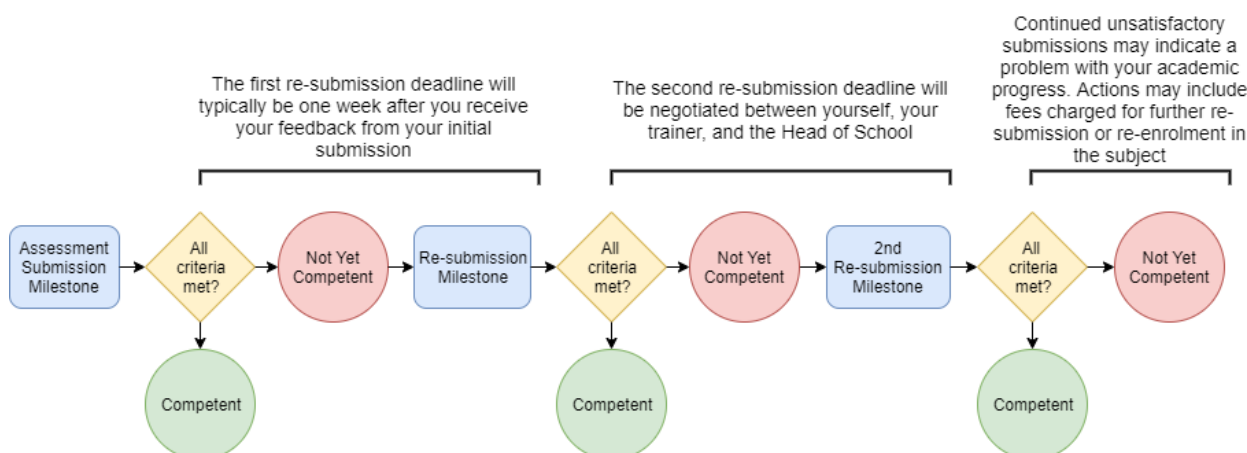
Formative assessment takes place as your teacher observes the development of your work throughout the subject and, although the assessor is likely to be aware of the evidence you are submitting, it is your responsibility to be prepared for the interview where a competency judgement is made (summative assessment). Forgetting something, or making a small mistake at the time of the milestone assessment, can be corrected. However, the assessor may choose to assess other candidates who are better prepared and return to you if time permits.

Upon completion of the assessment you will be issued with feedback and a record of the summative assessment and acknowledge that you have received the result. If you are absent for the nominated assessment milestone (without prior agreement or a sufficiently documented reason) you will be assessed as not yet competent.

## GRADING

The assessment you are undertaking will be graded as either *competent* or *not yet competent*.

## REASSESSMENT PROCESS



If you are assessed as being not yet competent you will receive clear, written and oral feedback on what you will need to do to achieve competence. Failing to submit an assessment will result in you being assessed as not yet competent. You will be given a reassessment milestone no more than one (1) week later to prepare your evidence. If you are unsuccessful after your reassessment, you may be asked to attend a meeting with your Head of School to discuss your progress or any support you may need and further opportunities to gain competency.

## REASONABLE ADJUSTMENTS

We recognise the need to make reasonable adjustments within our assessment and learning environments to meet your individual needs. If you need to speak confidentially to someone about your individual needs, please contact your teacher.

## FURTHER INFORMATION

For further information about assessment and support at AIE, please refer to the assessment and course progress sections of your student handbook.

# Software

## Core

### Microsoft Visual Studio

Microsoft's Visual Studio is the recommended IDE for this subject. Other IDEs may be employed if desired as the content of this subject is designed to be cross-platform and IDE agnostic, however we cannot guarantee that all subject material will operate as intended on other IDEs and platforms.

- <https://www.visualstudio.com/>



### GitKraken

GitKraken is a leading Git GUI client for Windows, Mac and Linux, used to create and maintain version control repositories. It helps developers become more productive with Git, and provides an integrated conflict editor, built-in code editor and task tracking. Other version control clients may be employed if desired.

- <https://www.gitkraken.com/>

### 7zip

7-Zip is a free and open-source file archiver, a utility used to place groups of files within compressed containers known as “archives”. This utility program will be necessary to package your assessment files for submission.

- <https://www.7-zip.org/download.html>

### Microsoft Word

Microsoft Word is industry-standard word processing software developed by Microsoft and used throughout the course for creating documents and reports. Microsoft Word allows documents to be saved in *word* format, as well as several other standard document formats, including *pdf*.

Learners will have access to Microsoft Word on campus but may also use alternate word processing software capable of loading and saving documents in *word* or *pdf* format.

- <https://www.microsoft.com/en-us/education/products/office/default.aspx>
- <https://www.openoffice.org/>
- <https://www.google.com.au/docs/about/>

## Suggested

### TortoiseGit

TortoiseGit is a Git revision control client, implemented as a Windows shell extension, allowing you to see the status of your Git repository and perform basic Git commands directly in your Windows file system.

- <https://tortoisegit.org/>

### WinMerge

WinMerge is a free software tool for data comparison and merging of text-like files. It is useful for determining what has changed between versions, and then merging changes between versions. It is an essential tool when working with version control systems to resolve conflicts.

- <https://winmerge.org/?lang=en>