



College of Engineering, Construction and Living Sciences  
Bachelor of Information Technology  
ID511001: Programming 2  
Level 5, Credits 15  
**Classroom Tasks**

## Assessment Overview

In this assessment, you will **unit test** your student management system **Windows Form Application** using **C#**.

## Learning Outcomes

At the successful completion of this course, learners will be able to:

1. Build interactive, event-driven GUI applications using pre-built components.
2. Declare and implement user-defined classes using encapsulation, inheritance and polymorphism.

## Assessments

Assessment	Weighting	Due Date	Learning Outcomes
Project 1	25%	21-06-2024 (Friday at 4.59 PM)	1 and 2
Project 2	35%	03-05-2024 (Friday at 4.59 PM)	1 and 2
Theory Examination	30%	26-06-2024 (Wednesday at 3.00 PM)	1 and 2
Classroom Tasks	10%	03-05-2024 (Friday at 4.59 PM)	1 and 2

## Conditions of Assessment

You will complete this assessment during your learner-managed time. However, there will be time during class to discuss the requirements and your progress on this assessment. This assessment will need to be completed by **Friday, 03 May 2024 at 4.59 PM**.

## Pass Criteria

This assessment is criterion-referenced (CRA) with a cumulative pass mark of **50%** over all assessments in **ID511001: Programming 2**.

## Authenticity

All parts of your submitted assessment **must** be completely your work. Do your best to complete this assessment without using an **AI generative tool**. You need to demonstrate to the course lecturer that you can meet the learning outcome(s) for this assessment.

However, if you get stuck, you can use an **AI generative tool** to help you get unstuck, permitting you to acknowledge that you have used it. In the assessment's repository **README.md** file, please include what prompt(s) you provided to the **AI generative tool** and how you used the response(s) to help you with your work. It also applies to code snippets retrieved from **StackOverflow** and **GitHub**.

Failure to do this may result in a mark of **zero** for this assessment.

## Policy on Submissions, Extensions, Resubmissions and Resits

The school's process concerning submissions, extensions, resubmissions and resits complies with **Otago Polytechnic | Te Pūkenga** policies. Learners can view policies on the **Otago Polytechnic | Te Pūkenga** website located at <https://www.op.ac.nz/about-us/governance-and-management/policies>.

## Submission

You **must** submit all application files via **GitHub Classroom**. Here is the URL to the repository you will use for your submission – <https://classroom.github.com/a/Wx7UHym1>. If you do not have not one, create a **.gitignore** and add the ignored files in this resource - <https://raw.githubusercontent.com/github/gitignore/main/VisualStudio.gitignore>. Create a branch called **classroom-tasks**. The latest application files in the **classroom-tasks** branch will be used to mark against the **Functionality** criterion. Please test before you submit. Partial marks **will not** be given for incomplete functionality. Late submissions will incur a **10% penalty per day**, rolling over at **5:00 PM**.

## Extensions

Familiarise yourself with the assessment due date. Extensions will **only** be granted if you are unable to complete the assessment by the due date because of **unforeseen circumstances outside your control**. The length of the extension granted will depend on the circumstances and **must** be negotiated with the course lecturer before the assessment due date. A medical certificate or support letter may be needed. Extensions will not be granted for poor time management or pressure of other assessments.

## Resubmissions

Learners may be requested to resubmit an assessment following a rework of part/s of the original assessment. Resubmissions are to be completed within a negotiable short time frame and usually **must** be completed within the timing of the course to which the assessment relates. Resubmissions will be available to learners who have made a genuine attempt at the first assessment opportunity and achieved a **D grade (40-49%)**. The maximum grade awarded for resubmission will be **C-**.

## Resits

Resits and reassessments **are not** applicable in **ID511001: Programming 2**.

## Instructions

You will need to submit an application and documentation that meet the following requirements:

### Functionality - Learning Outcome 1 (50%)

- The application needs to open without code or file structure modification in **Visual Studio**.
- You need to create 20 **unit tests** covering the following:
  - All **properties** in the **Institution** class. **Three** tests are expected.
  - All **properties** in the **Person** class. **Three** tests are expected.
  - All **methods** in the **CourseAssessmentMark** class. **Seven** tests are expected.
  - The number of **Institution**, **Department** and **Course** objects after seeding. **Three** tests are expected.
  - The salary of a **Lecturer**, **Senior Lecturer**, **Principal Lecturer** and **Associate Professor**. **Four** tests are expected.

### Code Quality and Best Practices - Learning Outcome 2 (45%)

- Appropriate naming of files, variables, methods and classes.
- Idiomatic use of values, control flow, data structures and in-built functions.
- Efficient algorithmic approach.
- Sufficient modularity.
- Each file has an **XML documentation comment** located at the top of the file. In the **assessment** directory of the **course materials** repository, you will find an **XML documentation comment** example in the **xml-documentation-comment.txt** file.
- Formatted code.
- No dead or unused code.

### Documentation and Git Usage - Learning Outcome 2 (5%)

- Provide the following in your repository **README.md** file:
  - How to run your unit tests?

### Additional Information

- **Do not** rewrite your **Git** history. It is important that the course lecturer can see how you worked on your assessment over time.