UNSW Course Outline

# COMP6771 Advanced C++ Programming - 2024

Published on the 03 Jun 2024

## General Course Information

**Course Code :** COMP6771
**Year :** 2024
**Term :** Term 2
**Teaching Period :** T2
**Is a multi-term course? :** No
**Faculty :** Faculty of Engineering
**Academic Unit :** School of Computer Science and Engineering
**Delivery Mode :** Multimodal
**Delivery Format :** Standard
**Delivery Location :** Kensington
**Campus :** Sydney
**Study Level :** Postgraduate, Undergraduate
**Units of Credit :** 6

**Useful Links**

Handbook Class Timetable

## Course Details & Outcomes

### Course Description

COMP6771 is an advanced programming course teaching practical aspects of intermediate/ advanced C++ programming. The course focuses on teaching the fundamentals of C++, followed by exploring powerful abstractions that C++ enables. This course focuses on using abstractions

as well as building abstractions.

COMP6771 is focused on modern, practical programming methods and tools. This course is designed for latter year CSE students with a reasonable degree of programming competencies.

The course is heavily supported by Christopher Di Bella , a UNSW CSE graduate who is a well-regarded expert on C++. His knowledge and expertise assists in forming and updating the course.

Our aim for students who complete this course satisfactorily is that they are highly competent in understanding C++ and it's core features, being able to build complex programs, data structures, and algorithms with C++, and being ready to immediately move into the workforce in areas that rely heavily on C++.

COMP6771 can be a challenging course for students due to the volume of work to complete in a 10 week period.

## Course Aims

This course aims to introduce the students to intermediate and advanced programming using C++. Students will be exposed to the core features of advance C++ with emphasis on how to incorporate them into well designed C++ programs. The first part of the course will introduce the basic features of C++, including references, functions, classes, C++ STL, operator overloading, and copy control. The second part will focus on more advanced topics such as exception handling, memory management, templates and generic programming, C++ template metaprogramming, input/output with C++ iostreams, C++ and efficiency issues, RTTI and effective C++ design guidelines. The structure, principles, and fundamental uses of the C++ Standard Library (including the Standard Template Library) will be introduced.

The course will be useful to those who wish to get a strong understanding of practical aspects of C++.

# Course Learning Outcomes

| Course Learning Outcomes |
|---|
| CLO1 : Design, build, and test C++ programs |
| CLO2 : Use abstractions (data structures, algorithms) to solve problems efficiently |
| CLO3 : Build abstractions (data structures, algorithms) to solve problems efficiently |
| CLO4 : Distinguish good, modern, widely-used practices from more outdated practices |
| CLO5 : Set up build and testing environments for C++ programs |
| CLO6 : Use advanced C++ features in program development. |

| Course Learning Outcomes | Assessment Item |
|---|---|
| CLO1 : Design, build, and test C++ programs | • Assignment 1<br>• Assignment 2<br>• Assignment 3<br>• Exam |
| CLO2 : Use abstractions (data structures, algorithms) to solve problems efficiently | • Assignment 1<br>• Assignment 2<br>• Exam |
| CLO3 : Build abstractions (data structures, algorithms) to solve problems efficiently | • Assignment 1<br>• Assignment 2<br>• Exam |
| CLO4 : Distinguish good, modern, widely-used practices from more outdated practices | • Assignment 3<br>• Assignment 1<br>• Assignment 2<br>• Exam |
| CLO5 : Set up build and testing environments for C++ programs | • Assignment 3<br>• Assignment 1<br>• Assignment 2 |
| CLO6 : Use advanced C++ features in program development. | • Assignment 3<br>• Assignment 2 |

# Learning and Teaching Technologies

https://cgi.cse.unsw.edu.au

# Assessments

## Assessment Structure

| Assessment Item | Weight | Relevant Dates |
|---|---|---|
| Assignment 1<br>Assessment Format: Individual | 20% | Due Date: Week 3 |
| Assignment 2<br>Assessment Format: Individual | 25% | Due Date: Week 7 |
| Assignment 3<br>Assessment Format: Individual | 30% | Due Date: Week 10 |
| Exam<br>Assessment Format: Individual | 25% | Due Date: Exam Period |

## Assessment Details

### Assignment 1

#### Assessment Overview

In Week 2 we learn about C++ libraries, and this assignment is your chance to practice those skills. In the role of client, the low-level details have already been dealt with, and student can focus their attention on solving more pressing problems. Having a library of thoroughly tested and reviewed types designed by field experts vastly broadens the kinds of tasks you can “easily” tackle. In this first assignment, student write the back-end for a program that heavily leverages the standard library to do nifty things.

This assignment has several purposes:

1. To explore C++’s value-semantics.
2. To stress the notion of abstraction as a mechanism for managing data and providing functionality without revealing the implementation.
3. To become more familiar with C++'s type system.
4. To gain familiarity with the C++ standard library

Assignments are auto-marked, with feedback on programming style given online by the tutors.

#### Course Learning Outcomes

- CLO1 : Design, build, and test C++ programs
- CLO2 : Use abstractions (data structures, algorithms) to solve problems efficiently
- CLO3 : Build abstractions (data structures, algorithms) to solve problems efficiently
- CLO4 : Distinguish good, modern, widely-used practices from more outdated practices

- CLO5 : Set up build and testing environments for C++ programs

Detailed Assessment Description

**Please refer to cgi.cse.unsw.edu.au for the correct Assessment and Weighting.**

## Assignment 2

Assessment Overview

Student will be writing copy and move constructors/assignment, member functions, non-member function spaceship operator, bidirectional iterator for user defined class.This means that the standard begin(), end(), cbegin(), cend(), rbegin(), rend(), crbegin(), crend() suite of functions need to be implemented as member functions.

Student must make sure their code does not perform its operations in a grossly inefficient manner. When testing, we will put a one second time limit on each test.

Assignments are auto-marked, with feedback on programming style given online by the tutors.

Course Learning Outcomes

- CLO1 : Design, build, and test C++ programs
- CLO2 : Use abstractions (data structures, algorithms) to solve problems efficiently
- CLO3 : Build abstractions (data structures, algorithms) to solve problems efficiently
- CLO4 : Distinguish good, modern, widely-used practices from more outdated practices
- CLO5 : Set up build and testing environments for C++ programs
- CLO6 : Use advanced C++ features in program development.

Detailed Assessment Description

**Please refer to cgi.cse.unsw.edu.au for the correct Assessment and Weighting.**

## Assignment 3

Assessment Overview

In this assignment, student will be implementing a simplified dataflow pipeline library, making use of both static polymorphism (templates) and dynamic polymorphism (virtual functions and inheritance). Users can then define their own types to be used with this library to perform useful computations.

Assignments are auto-marked, with feedback on programming style given online by the tutors.

Course Learning Outcomes

- CLO1 : Design, build, and test C++ programs
- CLO4 : Distinguish good, modern, widely-used practices from more outdated practices

- CLO5 : Set up build and testing environments for C++ programs
- CLO6 : Use advanced C++ features in program development.

Detailed Assessment Description

**Please refer to cgi.cse.unsw.edu.au for the correct Assessment and Weighting.**

### Exam

Assessment Overview

The exam involves solving practical problems in the CSE labs, during an invigilated session in the UNSW exam period. Questions are primarily auto-marked, with manual checking of "almost correct" solutions.

Course Learning Outcomes

- CLO1 : Design, build, and test C++ programs
- CLO2 : Use abstractions (data structures, algorithms) to solve problems efficiently
- CLO3 : Build abstractions (data structures, algorithms) to solve problems efficiently
- CLO4 : Distinguish good, modern, widely-used practices from more outdated practices

Detailed Assessment Description

**Please refer to cgi.cse.unsw.edu.au for the correct Assessment and Weighting.**

Hurdle rules

50% scaled score reached

## General Assessment Information

Grading Basis

Standard

# Course Schedule

| Teaching Week/Module | Activity Type | Content |
|---|---|---|
| Week 1 : 27 May - 2 June | Lecture | • Welcome & Getting Started<br>• C++ Basics<br>• Assignment 1 |
| Week 2 : 3 June - 9 June | Lecture | • STL Containers<br>• STL Iterators<br>• STL Algorithms |
| Week 3 : 10 June - 16 June | Lecture | • Class Types |
| Week 4 : 17 June - 23 June | Lecture | • Operator Overloads<br>• Custom Iterators<br>• Assignment 2 |
| Week 5 : 24 June - 30 June | Lecture | • Exceptions<br>• Resource Managements |
| Week 6 : 1 July - 7 July | Other | Flexibility Week |
| Week 7 : 8 July - 14 July | Lecture | • Dynamic Polymorphism |
| Week 8 : 15 July - 21 July | Lecture | • Templates<br>• Assignment 3 |
| Week 9 : 22 July - 28 July | Lecture | • Metaprogramming |
| Week 10 : 29 July - 4 August | Lecture | • Exam<br>• Guest Lectures |

## Attendance Requirements

Students are strongly encouraged to attend all classes and review lecture recordings.

## General Schedule Information

The timetable for this course is outlined clearly in the timetable for lectures, tutorials, and help sessions.

# Course Resources

## Prescribed Resources

Before commencing this course, students should:

- Be competent in constructing and designing programs in the language C (from COMP1511/1917 or equivalent)
- Be competent in understanding object-oriented (OO) programming methods (from COMP2511)
- Be competent with the basics of git usage (pull, push, add, commit)

We will spend minimum time covering basics of C and OO such as pointers, pointer arithmetic, classes, objects, and memory.

## Recommended Resources

There is no single text book that covers all of the material in this course at the right level of detail and using the same technology base as we are. The lectures should provide sufficient detail to introduce topics, and you will then study them in further depth in the tutorials, exercises and assignments. For some lectures, further reading material may be given for students who wish to gain a deeper understanding.

## Course Evaluation and Development

This course is evaluated each session using the MyExperience system.

This is being addressed during $24T2.

- Simplication of assessment specifications
- Simplication of environment setup
- Simplication of exercise structure
- Introduction of in-person open-book exam
- Refresh of style of all lecture content

# Staff Details

| Position | Name | Email | Location | Phone | Availability | Equitable Learning Services Contact | Primary Contact |
|---|---|---|---|---|---|---|---|
| Convenor | Hayden Smith | | | | | Yes | Yes |

# Other Useful Information

### Academic Information

#### I. Special consideration and supplementary assessment

If you have experienced an illness or misadventure beyond your control that will interfere with your assessment performance, you are eligible to apply for Special Consideration prior to, or within 3 working days of, submitting an assessment or sitting an exam.

Please note that UNSW has a Fit to Sit rule, which means that if you sit an exam, you are declaring yourself fit enough to do so and cannot later apply for Special Consideration.

For details of applying for Special Consideration and conditions for the award of supplementary assessment, please see the information on UNSW's Special Consideration page.

**II. Administrative matters and links**

All students are expected to read and be familiar with UNSW guidelines and polices. In particular, students should be familiar with the following:

- Attendance
- UNSW Email Address
- Special Consideration
- Exams
- Approved Calculators
- Academic Honesty and Plagiarism
- Equitable Learning Services

**III. Equity and diversity**

Those students who have a disability that requires some adjustment in their teaching or learning environment are encouraged to discuss their study needs with the course convener prior to, or at the commencement of, their course, or with the Equity Officer (Disability) in the Equitable Learning Services. Issues to be discussed may include access to materials, signers or note-takers, the provision of services and additional exam and assessment arrangements. Early notification is essential to enable any necessary adjustments to be made.

**IV. Professional Outcomes and Program Design**

Students are able to review the relevant professional outcomes and program designs for their streams by going to the following link: https://www.unsw.edu.au/engineering/student-life/student-resources/program-design.

*Note: This course outline sets out the description of classes at the date the Course Outline is published. The nature of classes may change during the Term after the Course Outline is published. Moodle or your primary learning management system (LMS) should be consulted for the up-to-date class descriptions. If there is any inconsistency in the description of activities between the University timetable and the Course Outline/Moodle/LMS, the description in the Course Outline/Moodle/LMS applies.*

## Academic Honesty and Plagarism

UNSW has an ongoing commitment to fostering a culture of learning informed by academic integrity. All UNSW students have a responsibility to adhere to this principle of academic integrity. Plagiarism undermines academic integrity and is not tolerated at UNSW. *Plagiarism at*

*UNSW is defined as using the words or ideas of others and passing them off as your own.*

Plagiarism is a type of intellectual theft. It can take many forms, from deliberate cheating to accidentally copying from a source without acknowledgement. UNSW has produced a website with a wealth of resources to support students to understand and avoid plagiarism, visit: student.unsw.edu.au/plagiarism. The Learning Centre assists students with understanding academic integrity and how not to plagiarise. They also hold workshops and can help students one-on-one.

You are also reminded that careful time management is an important part of study and one of the identified causes of plagiarism is poor time management. Students should allow sufficient time for research, drafting and the proper referencing of sources in preparing all assessment tasks.

Repeated plagiarism (even in first year), plagiarism after first year, or serious instances, may also be investigated under the Student Misconduct Procedures. The penalties under the procedures can include a reduction in marks, failing a course or for the most serious matters (like plagiarism in an honours thesis or contract cheating) even suspension from the university. The Student Misconduct Procedures are available here:

www.gs.unsw.edu.au/policy/documents/studentmisconductprocedures.pdf

## Submission of Assessment Tasks

Work submitted late without an approved extension by the course coordinator or delegated authority is subject to a late penalty of five percent (5%) of the maximum mark possible for that assessment item, per calendar day.

The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. This is for all assessments where a penalty applies.

Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

For some assessment items, a late penalty may not be appropriate. These will be clearly indicated in the course outline, and such assessments will receive a mark of zero if not completed by the specified date. Examples include:

- Weekly online tests or laboratory work worth a small proportion of the subject mark;
- Exams, peer feedback and team evaluation surveys;
- Online quizzes where answers are released to students on completion;
- Professional assessment tasks, where the intention is to create an authentic assessment that has an absolute submission date; and,
- Pass/Fail assessment tasks.

## Faculty-specific Information

Engineering Student Support Services – The Nucleus - enrolment, progression checks, clash requests, course issues or program-related queries

Engineering Industrial Training – Industrial training questions

UNSW Study Abroad – study abroad student enquiries (for inbound students)

UNSW Exchange – student exchange enquiries (for inbound students)

UNSW Future Students – potential student enquiries e.g. admissions, fees, programs, credit transfer

**Phone**

(+61 2) 9385 8500 – Nucleus Student Hub

(+61 2) 9385 7661 – Engineering Industrial Training

(+61 2) 9385 3179 – UNSW Study Abroad and UNSW Exchange (for inbound students)

## School Contact Information

**CSE Help! - on the Ground Floor of K17**

- For assistance with coursework assessments.

**The Nucleus Student Hub** - https://nucleus.unsw.edu.au/en/contact-us

- Course enrolment queries.

**Grievance Officer** - grievance-officer@cse.unsw.edu.au

- If the course convenor gives an inadequate response to a query or when the courses convenor does not respond to a query about assessment.

**Student Reps** - stureps@cse.unsw.edu.au

- If some aspect of a course needs urgent improvement. (e.g. Nobody responding to forum queries, cannot understand the lecturer)

You should **never** contact any of the following people directly:

- Vice Chancellor

- Pro-vice Chancellor Education (PVCE)

- Head of School

- CSE administrative staff

- CSE teaching support staff

They will simply bounce the email to one of the above, thereby creating an unnecessary level of indirection and a delay in the response.