



---

## Course Times and Locations

To be announced.

## Course Personnel

### Instructor:

To be announced.

### Instructional Support Assistant (ISA):

To be announced.

### Course Instructional Support Coordinator:

Karen Anderson  
MC 4010  
519-888-4567 x36656  
kaanders@uwaterloo.ca

## Calendar Description

Systematic methods for designing, coding, testing, and documenting medium-sized programs. Major topics include abstraction, modularity, software modeling, object-oriented programming and design, generic programming, testing and debugging.

## Required Background

- Experience programming with C/C++ (structures, strings, procedural abstractions, pointers, addresses, recursion, objects, inheritance, polymorphism), as covered and used in CS 137/138 and CS 241.
- Knowledge of container Abstract Data Types (lists, stacks, queues, trees), as covered and used in CS 138 and CS 241
- Ability to write functional specifications (pre / post conditions), as covered and used in CS 138

## Overall Goals

An essential goal of this course is to improve students' proficiency in designing and implementing modest-sized programs (5-10 modules). Course concepts that aim to fulfill this goal include programming abstractions, design principles, incremental development methods, debugging strategies, libraries, and software tools.

The emphasis is on (1) decomposing a program into multiple modules each of which can developed separately, and (2) using abstractions (interfaces, objects, composite ADTs) to reduce and simplify what a programmer needs to know about the rest of the program in order to implement a single module. Thus, the design of a program is evaluated not only on its correctness and performance, but also on its modularity, robustness, and generality. In addition to designing their own software solutions, students are asked to justify their design decisions and to evaluate others' designs.

## Learning Outcomes

- Design and implement data abstractions (ADTs, polymorphic objects, generic functions) in C++
- Critique designs with regards to cohesion, coupling, generality, robustness, information hiding
- Create software designs that are modular, general, robust, flexible
- Express software designs and behaviour using basic UML models:
  - Express the structure of an OO program as a class model
  - Express the state of a dynamic data structure as an object model
  - Express interactions between objects using a sequence diagram

- Develop and test programs incrementally
- Test and debug programs systematically
- Use simple software tools effectively, including makefiles, gdb, version control.

## Topic Overview

### Module Design (12 hours)

ADT design, function and operator overloading, modules and interfaces, namespaces, interface specification.

### Object-Oriented Design and Programming (12 hours)

Composite objects, OO design principles, OO design patterns, multiple inheritance and mixins

### Software Engineering and Tools (5 hours)

UML modelling, incremental development (make), testing and debugging (gdb), version control (e.g., svn)

### Generic Programming (5 hours)

STL algorithms, iterators

## Texts

Bruce Eckel, Thinking in C++ Volume 1, 2nd edition, Prentice Hall, 2000

Bruce Eckel, Chuck Allison, Thinking in C++ Volume 2, Prentice Hall, 2003

## Course Resources

**Course web page:** <http://www.student.cs.uwaterloo.ca/~cs247>

**Newsgroup:** [piazza.com/uwaterloo.ca/spring2015/cs247](http://piazza.com/uwaterloo.ca/spring2015/cs247)

**Course account:** [cs247@uwaterloo.ca](mailto:cs247@uwaterloo.ca)

## Evaluation

### Assignments (15%)

There will be three assignments, each consisting of short programming problems, design exercises, or software engineering tasks. No group work is allowed on assignments.

### Project (15%)

There will be a course project that can be done in teams of one or two students. The project is divided into two deliverables.

### Midterm (20%)

The midterm is worth 20% of your final grade

### Final Exam (50%)

The final exam is worth 50% of your final grade. ***You must achieve a passing grade on the exam component of the course (the midterm and final exam, weighted accordingly) in order to pass the course.*** Otherwise, your final grade will be the minimum of the computed final grade and the exam component grade.

## Course Policies

### Group Work Policy

No group work is permitted on assignments. Students may work on the course project in teams of size 1 or 2.

### Collaboration Policy

Some degree of collaboration is beneficial: you can learn a lot from others; you can avoid getting stuck; and teaching someone else can be the best way to cement your own understanding of a difficult concept. That said, your understanding of the course material will be deeper and more ingrained if you solve problems on your own. Thus, we permit and encourage *discussion* but not outright *collaboration*

(except where explicitly permitted, such as teamwork on the course project).

Specifically, you are allowed (and even encouraged) to discuss course concepts, assignments, and projects with other CS247 students **but only under the following restrictions.**

- No materials you bring to or take away from such meetings may be incorporated into your solution. In particular, you may not bring any of your code or solutions to the meeting.
- You may only bring away information in your long-term memory. You must destroy any materials that you and others create during the meeting. Then, you must spend 30 minutes without thinking about CS247 (e.g., watching a mindless TV show). After that, you can use whatever you still remember.
- You must write up anything you submit on your own. Your code (which includes tests and documentation), problem answers, etc. must represent your own understanding, as explained solely by you.
- You may not view other people's code or solutions. You may not share any of your own code (including, as always, tests and documentation) with others, including bringing it to a meeting with others. You may not allow anyone except the course staff access to your CS247 course directory, and you must restrict access to any Git repositories containing your CS247 work, or any other location where you keep CS247 solutions. Don't post large amounts of your code (more than about 5 lines) to a public forum (i.e., in a public Piazza post, to a Facebook group, to reddit, etc.)
- You must give credit where credit is due. Within each assignment/project submission, list everyone with whom you've had substantive discussions. Likewise, if you obtained a key idea from some other resource, such as a textbook or a website, then you should credit it (e.g., in comments at the top of your program submission).
- You may not view and/or use any substantive material or solutions from similar assignments this term or previous terms at UW or elsewhere, including anywhere on the Internet, transcribing solutions from any other source, etc.

We will use technological and other means to detect cheating.

### Late Policy

Up to one assignment and one project deliverable may be handed in, without penalty, up to 48 hours late. This late policy is meant to cover all minor reasons for missing a due date (e.g., minor medical illness, conflict with another assignment, etc.). There is no special process for invoking the late policy -- simply hand in your assignment late.

Anything handed in after the late-policy due date without a pre-approved extension (see below) will receive a grade of 0. If more than one assignment/project deliverable is handed in after the late-policy due date, one of the late submissions will receive a grade of 0.

If you have a serious (multiple-day) illness or absence, then you need to consult with the instructor in advance to make alternative arrangements for fulfilling the assignment deliverables of the course.

### Assignment Submission and Pickup

All assignments in this course will be submitted electronically. Marked assignments will be handed back in tutorials. After that, arrangements must be made to collect them from the course tutor. Anything left unclaimed (including midterms) at the end of the term will be shredded.

### Re-marking Policy

**Assignments:** email the ISA (cs247 at uwaterloo.ca) with Subject: cs247 Ax Re-mark Request, replacing x as appropriate. Clearly state the questions you want re-marked and include any supporting evidence for your case. Requests that include code changes to fix failing test cases will be ignored.

**Midterm:** Email cs247@uwaterloo.ca with the subject "CS247 Midterm Remark Request", clearly stating the questions you want re-marked. Include any supporting evidence for your case.

**Deadline for all re-mark requests:** You have two weeks after handback to deliver the request to the

ISA. All requests will be processed after the deadline to ensure fairness and consistency in marking.

**Note:** We will examine your entire assignment/midterm when remarking it. It is possible that you will receive a lower mark than your current mark.

### Missing an Examination

The only valid excuse for missing the midterm is illness substantiated by a doctor's note. In such a case, the weight of the midterm is added to the weight of the final exam. The only valid excuse for missing the final exam is illness substantiated by a doctor's note. In both cases, you must present a valid doctor's note to the course Instructional Support Coordinator (Karen Anderson, [kaanders@uwaterloo.ca](mailto:kaanders@uwaterloo.ca), MC 4010, x36656) as soon as possible -- preferably before the exam.

In the case of a missed final, you will receive an INC in the course, and you will have to make up the incomplete by writing the CS246 final exam in the F16 term.

## Students with disabilities

[AccessAbility Services](#), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the office at the beginning of each academic term.

## Integrity and Discipline

### Academic Integrity

In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check <http://www.uwaterloo.ca/academicintegrity/> for more information.

### Grievance

A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, <http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm>. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

### Discipline

A student is expected to know what constitutes academic integrity <http://www.uwaterloo.ca/academicintegrity> to avoid committing an academic offence, and to take responsibility for his/her actions.

A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about "rules" for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean.

For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, <http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm>. For typical penalties check Guidelines for the Assessment of Penalties, <http://secretariat.uwaterloo.ca/guidelines/penaltyguidelines.htm>.

### Appeals

A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) <http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm>.