CS 2300: Advanced Programming in C++ (3 credits)

Course Information

Section A: Tuesdays and Thursdays 10:05-11:20am in Innovation E204
Section B: Tuesdays and Thursdays 1:15-2:30pm in Innovation E204

Instructor: Lisa Dion (she/her)
Email: lisa.dion@uvm.edu

Instructor Drop-In Hours: TBD in Innovation E314 and virtually or in-person by appt.

TA Drop-In Hours: Sundays, Tuesdays, and Thursdays 5-7pm in Votey 303
starting the second Tue of the semester

Course prerequisites: CS 2240 and WIL1

Course Overview

This project-based course is separated into four modules. You will learn the details of C++ through a mixture of guided and open-ended projects, oral/written assignments, and other assessments. An emphasis will be put on creativity and collaboration.

WIL2: Writing and Information Literacy Tier 2

Courses that fulfill WIL2 will help students gain familiarity and fluency with genres, conventions, and formats typical in a discipline or field as well as develop a deeper understanding of how knowledge is accessed, developed, and shared. WIL2 courses build on skills and processes introduced in FWIL but refined through the conventions and practices of the field or discipline, including writing appropriately for different purposes, audiences, and contexts; posing and pursuing questions using relevant, reliable, and useful information while integrating and documenting sources correctly; understanding and evaluating ideas and evidence in texts; and developing flexible writing processes, including planning, drafting, revising, and polishing.

Learning Objectives

- To master C++ programming.
- To expand programming knowledge to projects that require multiple languages.
- To explore event-based programming through C++ graphics projects.
- To develop the skills necessary to effectively collaborate in teams.
- To apply computer science-based approaches to reading, writing, and working with information, including being able to engage with debates or conversations central to the field and to communicate to a range of audiences, as appropriate for course materials and context.
- To develop the capacity to reflect on and improve flexible and iterative processes for writing, reading, and pursuing inquiry in computer science.
- To use information ethically and with integrity, in terms of evaluating information for reliability and usefulness, crediting information sources, and understanding how information is generated and accessed in computer science.

Grading

This course will use a point-based (instead of percentage-based) grading system.

Each of the four modules in the course will offer over 300 points for you to earn through a variety of gradable assignments (see below) and there will be an optional final project. This means there will be a total of about 1500 points available for you to earn over the semester.

Your final grade will be based on the total number of points you earn by the end of the semester on this scale:

Point Range	Corresponding Letter Grade
At or over 1,000	A+
950 - 999	A
900 - 949	A-
866 - 899	B+
833 - 865	B
800 - 832	B-

Point Range	Corresponding Letter Grade
766 - 799	C+
733 - 765	C
700 - 732	C-
666 - 699	D+
633 - 665	D
600 - 632	D-
Under 600	F

Assignment Type	Description	Points Possible	Number Available
Guided Project	Follow step-by-step instructions given in a README file and/or coded comments to complete a programming project.	20+	3 per module
Applied Project	A structured but flexible project that has you explore an application of a topic.	50+	1 per module
Open-Ended Project	Apply concepts from the module to a project of your choosing. Open-Ended Projects are intended to build on one another.	100+	1 per module
Written / Oral Assignment	In-class presentations or written assignments related to topics in class.	15-20	4 per module
In-class Activities	Collaborative assessments during class time	20	2 per module
Final Project	Self-designed final project	50-200	1 overall
Other	Surveys and whatever else the instructor decides to throw in the mix.	TBD	TBD

- You do not need to complete all the assignments offered to you. You can choose which ones you are most interested in.
- You are encouraged to submit an assignment even if it isn't complete because earning some points is better than no points.

Self-Regulation

The grading system in this course does not lend itself well to habitual procrastinators. Procrastination is not a failure of time management, it is a failure of self-regulation. Merriam-Webster defines self-regulation as "the bringing of oneself … into a state of order, method, or uniformity." It's what enables you to get started on a project before the stress level of a looming deadline gets too high; it what pushes you to persevere through buggy code; it's what helps you consider both short-term and long-term goals to best choose how to spend your time now. Self-regulation is going to be crucial in this course. It will be up to you to make these regular habits outside of class.

Course Materials

You are required to have access to a computer for this course.

Software:

- Git is required. It is free to install from here: https://git-scm.com/downloads
- CLion is the supported IDE for this course. It is free with an educational license (renewable once a year) which you can get here: https://www.jetbrains.com/community/education

Tools:

- Gradescope will be used for submitting assignments. You will be able to view your grades with feedback.
- Brightspace will be used to post course materials and announcements.

Recommended Textbook:

- *Professional C++* by Marc Gregoire, 6th Edition. ISBN: 978-1-394-19317-2

Class Policy

- Late / make-up work will only be accepted under extenuating circumstances. Contact the instructor if you know you can't make a deadline.
- Essays and concept presentations will require a list of works cited, and any programming concepts not taught in class that you use in applied or open-ended projects must be cited.
- Questions about grades must be emailed to the instructor within a week of the grade release.
- There will be a zero tolerance policy for any vulgarity in the course. This includes in any work handed in and anything stored in Git repositories, as well as language expressed during class. Your grade for an assignment may be held at zero points until the situation is resolved.

Important Dates

Monday, September 8 - Add/Drop Deadline
Monday, October 27 - Withdraw Deadline
Tuesday & Thursday, November 25 & 27 - No class (Thanksgiving Break)
Note: We will not meet during finals week.

Tentative Schedule

Module	Date	Topic
Module 1: Intro / C++ Syntax	8/26	Syllabus, overview of Module 1 assignments
	8/28	Review of GitHub setup, pointer activity
	9/2	Student presentations of key concepts
	9/4	Debugging activity
	9/9	Input validation activity
	9/11	Student demo presentations / module ending help
Module 2: Advanced OOP Concepts / Intro to Teamwork	9/16	Overview of Module 2 assignments
	9/18	Teamwork discussion, collaborating on git, merge conflicts activity
	9/23	Has-A and Is-A class relationships overview and activity
	9/25	Student presentations of key concepts
	9/30	Code test cases activity
	10/2	Student demo presentations / module ending help
Module 3: Multiple Programming Languages	10/7	Overview of Module 3 assignments, intro to CLI
	10/9	Command line and multiple programming languages activities
	10/14	Website activity
	10/16	Student CLI presentations
	10/21	Open source & advanced git activity
	10/23	Student demo presentations / module ending help
Module 4: Graphics	10/28	Overview of Module 4 assignments, intro to graphics
	10/30	Graphics starter activity
	11/4	Graphics drawing and listeners activity
	11/6	Graphics practice activities
	11/11	Advanced graphics activity
	11/13	Student demo presentations / module ending help
Final Project	11/18-12/4	Final project work time
	Finals week	Final project due (we will not meet during the finals week slot)

Academic Integrity

You must abide by the Code of Academic Integrity (https://www.uvm.edu/policies/code-academic-integrity). Any suspected violation will be reported to the Center for Student Conduct.

Note on Generative AI: you may use AI to enhance your learning. You may not use AI to generate, in whole or part, work that is submitted for grading.

Accommodations

In keeping with University policy, any student with a documented disability interested in utilizing ADA accommodations should contact Student Accessibility Services (SAS), the office of Disability Services on campus for students. SAS works with students and faculty in an interactive process to explore reasonable and appropriate accommodations, which are communicated to faculty in an accommodation letter. All students are strongly recommended to discuss with their faculty the accommodations they plan to use in each course. Faculty who receive Letters of Accommodation with Disability Related Flexible accommodations will need to fill out the Disability Related Flexibility Agreement. Any questions from faculty or students on the agreement should be directed to the SAS specialist who is indicated on the letter.

Religious Holidays

Students have the right to practice the religion of their choice. Each semester, students should submit in writing to their instructors as early as possible and at least one week before their documented religious holiday the date(s) of the conflict or absence.