
CS368 - C++ for Java Programmers

1 Credit – Spring 2020

Dr. Marc Renault

mrenault@cs.wisc.edu

Office: CS 6382

Hours: Tuesdays, 14:00 - 16:00

Section 002 Wed 9:55 - 10:45

Location: Noland 168

Section 003 Wed 14:25 - 15:15

Location: Grainger 2080

Calendar Description

For students interested in learning a particular programming language. Focuses on a specific language offered at one of three levels: beginner, intermediate, and advanced. Students may repeat the course if the topic title is different. Enroll Info: None

Course Aims and Outcomes

This course is at the intermediate level course and introduces the C++ programming language. The course assumes a working knowledge of Java and basic object oriented concepts.

Specific Learning Outcomes

Students successfully completing this course will be able to:

- Read and understand C++ code
- Write C++ programs making use of the abstractions provided by the language
- Understand the lower level details of memory management like pointers and references
- Organize and build multi-file projects using the make tool
- Solve real world programming problems using C++ as a tool

Texts

(These texts are optional)

- **www.cppreference.com** Online resource for C++ (and C) language.
- **Lippman, Lajoie, and Moo. *C++ Primer, 5th Edition*. Pearson, 2013.** Now with C++11! The classic (presumable because it was the textbook I used when learning C++) text for learning C++.
- **Stroustrup. *A Tour of C++* Pearson, 2018.** The creator of C++, Bjarne Stroustrup, gives an overview of the current (as of 2018) state of C++, i.e. C++17.
- **Kernighan and Ritchie. *C Programming Language, 2nd Edition*. Prentice Hall PTG, 1988.** This is the standard C Programming reference used the world over (the C programming course I taught at Université Paris Diderot - Paris 7 used this book). Why? Because sometimes C++ programmers just need a good C reference book.

Format and Procedures

Class Structure

The classes will be a combination of lectures, discussions and exercises. You are encouraged to ask questions throughout the class and to be engaged in the discussions. During the exercises, please feel free to engage with your peers or myself as you work on the problems. Finally, please keep all your devices on silent during the class.

Note that this class will be Credit / No Credit (Cr / N).

Participation

In order to facility discussions and to determine participation grades, we will use TopHat in this course. The TopHat questions will be graded on participation not correctness. The participation score will be based on the number of questions answered during a lecture which will be averaged over all the lectures.

Assignment Submissions

Quizzes: A portion of your final grade (21%) will be based on a Canvas quiz that must be done prior to submitting the assignment. There are not late penalties for the Assignment quizzes. No marks will be given for quizzes that are not completed by May 1.

Code: The code should be logically organized into files and include a markdown `README.md` (see <https://guides.github.com/features/mastering-markdown/>). The `README.md` should give a brief description of the program, the organization of the code, how to compile the code, and how to run the program.

There must be one or more `makefile(s)`, and the code should compile by issuing the `make(1)` command. No marks for functionality will be awarded for code that does not compile correctly.

Late Submissions: A 10% deduction will be applied per week that the submission (with the penalty occuring the first second it is late). That is, immediately after the deadline would be a 10% deduction. The next day would also be a 10% deduction. Seven days late plus 1 second would be a 20% deduction, etc.

No submissions for any assignments will be accepted after May 1.

Citations: Regardless of the level of help provided, you should cite all the sources (websites, people, etc) that you used in completing the assignment.

Piazza

Both sections will use the same Piazza message board. You can sign up for the board at: <https://piazza.com/wisc/spring2020/cs368>.

Course Requirements

The course requirements are broken practical components (Assignments), theoretical components (Quizzes) and participation components (TopHat). Students who achieve a total overall grade of 70% or greater will receive a grade of Cr.

Assignments (70%)

The practical component consists of seven (7) programming assignments. Each assignment is worth 10%. (For late policy, see Assignment Submissions section above.)

Prior to submitting an assignment, you will have to complete the related quiz.

You are encouraged to begin the assignment early to give yourself enough time to plan, code and test. If you have any questions on the assignments, please ask me, a TA, or post on Piazza as early as possible.

Quizzes (21%)

Associated with each assignment is a Canvas quiz that must be completed before submitting the assignment to Gradescope. (For late policy, see Assignment Submissions section above.)

TopHat (9%)

During each lecture, there will be some TopHat questions which will be used to calculate a participation grade. Each question in a lecture will have the same weight and be grade based on participation (not correctness). Your grade for a given lecture will be the number of questions answered out of the number of questions asked. Your overall all grade will be an average of all the lecture TopHat grades.

If there is a lecture with not TopHat questions, it will not be used in the calculation of the TopHat grade. If your TopHat score is 80% or higher, you will be credited the full 9 points. If it is less than 80%, you will be credited $9x$ points, where x is your TopHat percentage score. I.e., a TopHat score of 79% would result in 7.11 out of 9 points.

Assignment Due Dates

Assignment	Due Date	Points
Assignment 1 – Basics	Feb 7	10%
Assignment 2 – STL	Feb 21	10%
Assignment 3 – File I/O	Feb 28	10%
Assignment 4 – OO	Mar 13	10%
Assignment 5 – Overloading operators	Apr 3	10%
Assignment 6 – Generics	Apr 17	10%
Assignment 7 – Free for all	May 1	10%

Schedule

(This schedule is subject to change.)

Week	Date	Topic (tentative)
1	Jan 22	Course Intro, Basic C++, Strings, I/O, and g++
2	Jan 29	More basics: Header Files, make, preprocessing, and functions
3	Feb 5	Pointers, references and memory
4	Feb 12	Basic containers (STL)
5	Feb 19	File I/O
6	Feb 26	Classes and the rule of 3
7	Mar 4	C++ Object oriented programming
8	Mar 11	Operator overloading

9	Mar 18	<i>Spring Break</i>
10	Mar 25	Generic programming
11	Apr 1	More containers and algorithms
12	Apr 8	Exception handling
13	Apr 14	Smart pointers
14	Apr 21	Other C++ (Regex and lambdas)
15	Apr 28	Other C++ (Random numbers, extern C)

A Note on Academic Dishonesty

More important than the exact grade you receive (especially in a Credit / No Credit course such as this), is the knowledge you gain. Specifically, you are here to learn C++. Don't cheat yourself by submitting other people's work.

University Policies

Rules, Rights and Responsibilities

See <http://guide.wisc.edu/undergraduate/#rulesrightsandresponsibilitiestext>.

Academic Integrity

By enrolling in this course, each student assumes the responsibilities of an active participant in UW-Madison's community of scholars in which everyone's academic work and behavior are held to the highest academic integrity standards. Academic misconduct compromises the integrity of the university. Cheating, fabrication, plagiarism, unauthorized collaboration, and helping others commit these acts are examples of academic misconduct, which can result in disciplinary action. This includes but is not limited to failure on the assignment/course, disciplinary probation, or suspension. Substantial or repeated cases of misconduct will be forwarded to the Office of Student Conduct & Community Standards for additional review. For more information, refer to <https://conduct.students.wisc.edu/academic-integrity/>.

Accommodations for Students with Disabilities

McBurney Disability Resource Center syllabus statement: "The University of Wisconsin-Madison supports the right of all enrolled students to a full and equal educational opportunity. The Americans with Disabilities Act (ADA), Wisconsin State Statute (36.12), and UW-Madison policy (Faculty Document 1071) require that students with disabilities be reasonably accommodated in instruction and campus life. Reasonable accommodations for students with disabilities is a shared faculty and student responsibility. Students are expected to inform faculty [me] of their need for instructional accommodations by the end of the third week of the semester, or as soon as possible after a disability has been incurred or recognized. Faculty [I], will work either directly with the student [you] or in coordination with the McBurney Center to identify and provide reasonable instructional accommodations. Disability information, including instructional accommodations as part of a student's educational record, is confidential and protected under FERPA." <http://mcburney.wisc.edu/facstaffother/faculty/syllabus.php>

Diversity and Inclusion

Institutional statement on diversity: "Diversity is a source of strength, creativity, and innovation for UW-Madison. We value the contributions of each person and respect the profound ways their identity, culture, background, experience, status, abilities, and opinion enrich the university community. We commit ourselves to the pursuit of excellence in teaching, research, outreach, and diversity as inextricably linked goals.

The University of Wisconsin-Madison fulfills its public mission by creating a welcoming and inclusive community for people from every background – people who as students, faculty, and staff serve Wisconsin and the world." <https://diversity.wisc.edu/>