
Course objective:  You will have the opportunity to learn the C++ programming language. We will cover the language rather quickly and you are expected to pursue learning the language on your own. The rest of the course is dedicated to exploring algorithm development methodologies. Becoming good at developing algorithms requires practice and patience, because almost always, the methodologies we study don’t work. They only give you insight into the problem you are studying. After successfully completing this course you will have a solid repertoire of tools for solving problems. These tools will be most useful in solving computing problems, but some are applicable to all areas of study and to life.

Grading for the course:  There will be 6-8 problem sets and multiple programming assignments for this course. As I create them, I will send out an email and you can find them on the course website, which also contains other information pertinent to the course, including a tentative schedule. The first midterm exam will be on Thursday, October 12, 2017, and the second on Thursday, November 9, 2017. The comprehensive final exam will be Friday, December 8, 2017 from 8:00 a.m.–10:00 a.m. Note that I consider the final an extremely important evaluative tool. If your score on the final exceeds the sum of your scores on the two midterms, I will replace the sum of your scores on the two midterms with your score on the final. I determine final grades on the basis of what you learn, so you must strive to learn the material as well as you possibly can. My performance, the difficulty of the exams, problem sets and programming assignments, and your performance all enter into the process of determining grades. However, if you get more than half of the total points, you will earn a C or higher. The A and B cutoffs are typically set at approximately 1/3 and 2/3 the distance between the high score and the C cutoff. I do not use the +/- grading system. I will clarify this process after each of the midterm exams and again before the final exam.

Course Requirements:  You must complete all problem sets, programming assignments, and the final exam before you will be eligible for a passing grade in this class. Note that completing all of the components does not guarantee a passing grade. Assignments that have not been turned in will be denoted by a “I” in the grade report. Late assignments are worth up to 50% of the original maximum. You will not pass this course if any “I’s” remain in your grade report at the beginning of the final exam. I will send you a grade report via email to your account on cs after each assignment and exam is graded.

\footnote{Copyright © 2017, Marty J. Wolf under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) license. See http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode for details.}
Each assignment is due at the beginning of the class on its due date. You can (and must if you wish to be eligible for a passing grade) turn them in after the due date/time. However, any work turned in after the due date and time will receive a score of at most half if I have begun the grading process.

You should begin each homework as it is handed out. You will not finish it if you begin it the day before it is due. By starting a homework early, you have the opportunity to ask me to clarify those things you do not understand. Also, there will be no make-up exams. Since the final exam is comprehensive, it will be used to compute a make up score for a missed exam.

If you are having difficulty with any aspect of this class please talk to me in a timely manner as more options are available to us if you approach me early. I will do my best to deal with your situation.

As a student in this class you are expected to spend a substantial amount of time learning and pursuing knowledge. Learning is not an easy process. As a guideline you are expected to attend class three hours per week and spend at least nine hours per week involved in out of class projects. This means an out of class commitment of 150 *quality* hours over the course over the next seventeen weeks. Plan to spend this much time!

It is essential that you commit yourself to regular studying. Without a consistent effort you will not meet deadlines and you will not find the assistance from me you need to complete problem sets and programming assignments.

Assignment guidelines: You can develop your programs in any environment that you choose, however, they must compile under g++ on the workstations in the CS Lab. As we move through the course you will gain familiarity with Linux, makefiles, g++ or any other system software needed for this course. Unless otherwise noted, each program must be completed exclusively by you.

Problem sets must be neat and readable. You can hand write them, but you should word process them when appropriate. I will not grade what I cannot read. Exams must also be written neatly. I will not score parts I cannot read.

Academic Honesty: Please be aware that the University’s policy for Academic Integrity appears in the Student Guide. I expect that each student has read this material. If you do not understand what is meant by this policy, or if you are confused by terms such as plagiarism, cheating, or collusion, please discuss this policy with me, your advisor, or another faculty member as soon as possible. I absolutely require that each student in this class fulfill his or her academic obligations in a fair and honest manner. This includes turning in work that is uniquely yours, unless I explicitly require you to work on a project in a group.

To this end, any two or more students who turn in work that looks even remotely similar will be given at most half credit for the work (i.e., I will grade it once and split the points among those students presenting similar work). I strongly suggest that if you work with others you only work together in the idea generation phase and that when it comes to writing or typing your work, you do so independently. It is in your best interest to never look at any solutions written by another student and to never let another students see any solutions you have written. If you do turn in work that I suspect is the result of cheating, I reserve the right to carry out the sanctions listed in the Student Guide.

Specific items that I consider cheating on programming assignments or problem sets are:
1. Turning in someone else’s work as your own (with or without that person’s consent). This includes turning in a copy of something that can be mechanically transformed into a copy of someone else’s work. Don’t even try to disguise cheating by simply modifying someone else’s work and calling it your own.

2. Allowing someone else to turn in your work as his or her own work. This includes allowing fellow students access to your electronic copy.

3. Using a solution developed by a student in a previous term.

4. Using a solution found in a book or journal article.

**Class participation:** Do not miss class. I try to present material differently than it is presented in the text in order to give you an additional point of view. You will find a tentative reading schedule at the web site listed above. I expect that you will have read the material for the day. You are responsible for all announcements and material covered in the event that you do miss class. I am not a source for that information. You must get it from one of your classmates.

In addition to attending class, you are required to participate in class. Participation involves following the lecture, asking questions and answering questions.

In the event of snow, I will hold class unless the university cancels classes. If you must travel, it is your responsibility to use good judgment as to whether to attend that day.

**Notice:** I would like to make sure that all the materials, discussions and activities that are part of the course are accessible to you. If you would like to request accommodations or other services, please contact me as soon as possible. It is also possible to contact Disability Services either in Decker Hall Room 202, by phone at 218 755-3883, via e-mail at Disabilityservices@bemidjistate.edu, or through the Minnesota Relay Service at 1-800-627-3529.