
Course objective: The purpose of this course is to bring together in a single project most of the fundamental aspects of computer science. The main task of this course is for you to design a (simple) programming language and implement a compiler for it. We will spend approximately six weeks studying the basic components of a compiler and the remainder of the semester designing a language and implementing a compiler for that language.

Grading for the course: There will be a few in class/homework assignments that align with the readings during the eight weeks of study. Shortly before Spring Break you will be assigned to a project team. After we define a language as a class, your team will implement a compiler for that language. There will be no exams.

Final grades will be a strict 90/80/70. The compiler project to constitute the bulk of your grade. Your compiler will be evaluated on

1. it’s structure and effectiveness,

2. the quality of the documentation, and

3. the quality and complexity of the test cases you produce

Each compiler team must arrange a meeting with the course instructor in the CS Lab (HS 104) to demonstrate their compiler. All members of the project team are expected to attend this meeting and each team member must be prepared to discuss her/his contributions to the compiler. This meeting can take place anytime after the team deems the compiler complete, but must be completed by 10am on Tuesday, May 1, 2018. The official final examination period for this class is 8am-10am on Tuesday, May 1, 2018. Note that you should schedule this meeting as soon as possible.

You will also be given individual score based on your contribution to your team’s compiler.

Since much of the class meeting time is dedicated to group work on your team’s compiler, you are required to attend class and be on time. If you are late more than six times on Project Days, the maximum grade you can earn is a B, regardless of the quality of your team’s compiler or your contribution to it. If you are late more than twelve times on Project Days, the maximum grade you can earn is a C, regardless of the quality of your team’s compiler or your contribution to it. Note that anytime you miss class or are late by more than 15 minutes on a Project Day, it will count as two late times for consideration for your highest grade.

Course Requirements: You must complete all of the in class/homework assignments before you will allowed to participate on a compiler team. Note that not each Project Day you are not participating on a compiler team counts a one day absent.

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

1Copyright © 2018, 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.
Assignment guidelines: All programming assignments must compile under g++ on the Linux workstations in the CS Lab. (Note that this does not mean the CS Server.) You must supply appropriate makefiles.

Academic Honesty: Please be aware that the University’s policy for Academic Integrity appears in the Student Guide. I expect that you have 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 you fulfill your 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 or on the web.

Class participation: In the event of snow, I will hold class unless the university cancels classes or closes. If you must travel, it is your responsibility to use good judgment as to whether to attend that day. Note that the late/absent policy identified above applies unless the university cancels classes.

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 Sanford Hall, 201, by phone at 218 755-3883, via e-mail at Disabilityservices@bemidjistate.edu, or through the Minnesota Relay Service at 1-800-627-3529.