Class schedule: Monday-Wednesday-Friday, 09:10am–10:00am


Instructor: Dr. Gabriel Dos Reis ([gdr@cs.tamu.edu](mailto:gdr@cs.tamu.edu))
- Office: 410C H. R. Bright Building
- Office hours: By appointment

Teaching Assistant: Tim Mann ([mann@cse.tamu.edu](mailto:mann@cse.tamu.edu))
- Office: 339 H. R. Bright Building
- Office hours: Monday-Thursday, 10:00am–11:15am

1 Topics and Goals

This course is intended as an intensive programming experience that integrates core concepts in Computer Science and familiarizes students with a variety of programming/development tools and techniques. Students will primarily work in small teams on month-long projects emphasizing different specializations within computer science. The course focuses on honing good programming techniques to ease code integration, reuse, and clarity.

The primary goal for this class is to have students emerge with strong programming skills, able to address both individual and team programming challenges competently. The class is meant to allow students to improve their programming skills through significant practice.

Among the topics to be covered in lecture periods are:
• Style considerations in writing code
• Design of software systems and APIs
• Coding beyond the single component
• Basic collaborative software coding practices
• Design for portability, performance, testability
• Specification and documentation
• Basic software tools and their use
• Subject-specific topics related to the team projects

Though many topics will overlap, this course is not intended to be as in-depth or comprehensive as a standard software engineering course, which focuses more on project management – students may take the software engineering class after taking this class.

Note: You should expect to spend a significant amount of time (>10 hours/week) outside of class time on programming projects. This may require meeting with team members outside of the class/lab periods.

2 Prerequisites

This class is intended for students who have completed CPSC 314 – Programming Languages, and are concurrently taking CPSC 313 – Intro to Computer Systems. It is meant to be somewhat of a “capstone” course for the lower-level computer science courses, before taking courses in the upper-level tracks.

3 Class Meetings

We will meet a total of 20 lecture periods over the course of the semester. The idea is to “front-load” these lectures in the earlier part of the semester,
to cover material that might be useful when working on the programming projects, and spend less lecture time during the project periods themselves. The list of days we will meet will be provided on the course web page.

There is a final exam time reserved for this class. Students should leave that time available until instructed otherwise, since it might be used for project presentations or something similar. However, there will not be a final exam in the course.

4 Grading

There will be three major projects in the course, each counting for 28% of the overall grade. Specific grading practices for each project will be announced when that project is given out, but the grade may include factors such as evaluation of code clarity, teamwork, etc. Peer evaluation may be used as a significant contributing factor to these grades. The remaining 16% of the grade will be an individual grade based on individual exercises, quizzes, participation in the course survey, and an evaluation of class participation (which might include participation in code reviews). Individual assignments will be small programming assignments to be completed on an individual basis.

The 16% of the grade will start off as being based totally on instructor judgement of class participation and effort. As the course progresses, any quizzes given out, individual assignments given out, or other specific graded material will note the portion of this individual grade which that quiz/assignment/etc. affects. The remainder of the individual grade will be based on the subjective class participation and effort grade. For example, if there are 8 quizzes at 1% each, one individual assignment at 4%, and participating in the course evaluation is 2%, then the remaining 2% is based on the subjective evaluation.

The grades will be assigned on absolute scale:

- A=90%–100%
- B=80%–89%
- C=70%–79%
- D=60%–69%
• F=0%–59%

• F*=plagiarism or cheating

In addition to this, the instructor reserves the right to provide a relative or absolute curve to the class. Also, the instructor may raise the grades of any students near a borderline based on a subjective evaluation of class participation and effort.

5 Policies

5.1 Attendance

Attendance is expected in the course, and may be recorded in both lectures and labs. 16% of the course grade will be based on individual evaluation of assignments and class participation, and repeated absences may negatively affect the grade. In addition, students might miss quizzes, which will not be made up without prior approval. Students with absences should notify the instructor ahead of time about any planned missed classes or labs.

5.2 Late Assignments

Each project will have a specified date and time at which it is due, and dates and times for which various intermediate parts of the project are due. Projects that are turned in late will have a penalty applied to the overall project grade, which will affect the grade given on that project for all team members (if individual reports are late, those will affect only the grade for that team member). The total number of minutes, \( m \), that assignments within a project are late will be added up. The final grade on the project will be multiplied by \( 0.9998^m \). For example, if the project is 1 hour late, you lose a bit over 1%. If it is one day late, you lose about 25%. After 3 days, you’re down to 42% of your grade lost.

5.3 Course Evaluation

An online course evaluation will be used for the class. Participation in this evaluation will affect the 16% portion of the course grade (the specific
amount will be announced toward the end of the course).

5.4 Communication

A class web page (listed at the top of this syllabus) will be maintained throughout the semester. Students are responsible for checking both the web page and email regularly for class updates.

5.5 Code Documentation

A key part of this class is understanding the importance of clear code construction and documentation. So, when assignments are graded, a significant portion of the grade may be based on an evaluation of how well the code is written, and how easy it is to follow. Just producing code that “works” is not sufficient; it will be your responsibility to produce code that the grader can follow.

5.6 Cell Phones

Any calls received during lecture will result in an automatic 3% penalty off your final course grade. Excuses will result in double the penalty. These points can be re-gained only by a written compensatory assignment of the instructor’s choosing, and only if it is your first offense.

6 Required Textbooks

We will be using the following textbook:


Other books that may be drawn from, and that might be useful references include both the first edition of *Code Complete*, as well as


- *Code Craft*, by Pete Goodliffe, No Starch, 2007. (Note: this book is available to read online for free through TAMU).
7 Academic Integrity Policy

An Aggie does not lie, cheat, or steal or tolerate those who do.

The Honor Council Rules and Procedures are available on the web

http://www.tamu.edu/aggiehonor

For this class, certain aspects of the honor code need to be clarified.

1. There may be times in this course where you or your team make use of external code/software/libraries. Whenever this is done, you must make sure that, in addition to following any restrictions on that code itself, you clearly document what the source of the external code was, and how it was used.

2. There may be cases in this course where you or your team seeks outside assistance related to one of the projects. Any assistance received from people other than members of your team, the professor, teaching assistant, or peer teacher needs to be clearly documented.

3. You will be working in team environments in this course, and your work as a team will be used to determine grades. As such, it is your responsibility, when asked, to:

   (a) accurately describe the work that you have done on a team project. Claiming credit for work that you have not done or that others did instead is a violation of the code.

   (b) accurately describe (to the best of your knowledge) the performance of other team members. “Covering” for another team member (claiming they did more work than you know they did) or “spiking” them (claiming they did less work than you know they did) are examples of honor code violations.

   (c) prevent (as best you can) or report (known) violations of the honor code by your other team members. You share responsibility when a project is turned in; if you are aware of a teammate having violated the code in his/her work on the project, and do not report it, you are claiming credit for that violation yourself.

If there are any questions or concerns about whether an action is appropriate, you should check with the professor or teaching assistant first. If in doubt, assume that it is not appropriate.
8 Americans with Disabilities Act (ADA) Policy

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact the Department of Student Life, Services for Students with Disabilities, in Cain Hall Room B118 of the Koldus Building or call 845-1637.