

Instructor

Dr. Kenneth W. Regan, 326 Davis Hall Hall, 645-4738 regan@buffalo.edu

TAs:

Mingxi Lei Davis 302 no phone mingxile@buffalo.edu
Yutian Yan Davis 302 no phone yutianya@buffalo.edu

GTAs:

Praneeth Sai Kotha Davis 302 no phone pkotha@buffalo.edu
Vindhya Nuthalapati Davis 302 no phone vindhyan@buffalo.edu

UTAs:

Sarthak Agarwal Davis 302 no phone sarthaka@buffalo.edu
Sean Grzenda Davis 302 no phone seangrze@buffalo.edu
Pantelis Kiamos Davis 302 no phone pskiamos@buffalo.edu
Korey Liu Davis 302 no phone koreyliu@buffalo.edu
Vrund Patel Davis 302 no phone vrundpat@buffalo.edu
Romika Sairam Davis 302 no phone romikasa@buffalo.edu

Lectures

LEC MWF 1–1:50pm in Knox 110

Recitations—All in the Bell 340 programming lab.

- (R1) Mon. 3:00–3:50pm
- (R2) Wed. 5:00–5:50pm
- (R3) Mon. 5:00–5:50pm
- (R4) Wed. 3:00–3:50pm
- (R5) Mon. 10:00–10:50am
- (R6) Tue. 1:00–1:50pm
- (R7) Mon. 2:00–2:50pm
- (R8) Wed. 6:00–6:50pm

Required Reading:

- (1) [LL17] Mark C. Lewis and Lisa L. Lacher *Object-Orientation, Abstraction, and Data Structures Using Scala (2nd ed.)*, CRC Press, 2017.
- (2) Web pages, maintained at www.cse.buffalo.edu/~regan/cse250/ and possibly at other locations, including sample code.

- (3) *Piazza*, which will be used for class discussions and time-sensitive announcements. Students are invited to post queries of general interest. Please do not, however, post answers unless and until cleared with the instructors and TAs.

The webpages will hold official information and handouts and items that tend not to change much over time, whereas the newsgroup will be the preferred vehicle for assistance with projects and homework. *Please do not print out copies of webpage documents* (unless instructed to do so), including my posted course notes.

Examinations:

- Two *prelims*—roughly 1–2 weeks past the 1/3 and 2/3 points of the course. The **First Prelim Exam** is set for **Wed., March 9** *in class period*.
- One *cumulative* 3-hr. final.
- Possible quiz components of assignments.

Grading: The course will be graded on a total-points system. Letter grades will also be given for individual exams and some assignments, as a help in telling you where you stand, but only the point totals will have official significance. The weighting of grades in this course will be:

Prelims:	$2 \times 10\% = 20\%$
Final:	30%
Homework:	50% (split between problem sets, projects; quizzes up to 5%)

Besides the “pop-quiz” option, we reserve the right of 5% leeway in the weights for assigning the final letter grade. This is typically done for students who do markedly well on the final exam—treating it as though it were weighted 35%. This will only be done to an individual student’s advantage, and will have no effect on others’ grades. It is predicated on recitation attendance.

Once all points are converted to percentages, the course will use a pre-set curve: 90% = A, 84 = A-, 78 = B+, 72 = B, 66 = B-, 60 = C+, 54 = C, 48 = C-, 42 = D+, 36 = D. Exams and assignments will be “curved” further only if some error or unforeseen circumstance affects the results.

1 Assignments

Problem sets will involve both pencil-and-paper questions and exercises that require programming (although in some cases, only “pseudocode” will be asked for). The 50% for assignments will be divided roughly 25% for homeworks and 25% for projects, but the line may not be so sharp—e.g. a “project” may have a hardcopy piece, and the first “homeworks” will ask you to submit some small Scala program(s).

Programming exercises may be developed on any system and IDE (such as IntelliJ or Eclipse) of your choosing, using Scala 2.13.x (*not Scala 3*). In the past, when compatibility was more an issue with C++ and its libraries, and with CSE250 being the first use of C++, we stated a rule that code had to be tested at the command line on the undergraduate CSE machine `timberlake`, which was used for automated grading. Now submissions will be using *CSE Autolab* (and a backup system if such becomes necessary). My understanding is that Scala is relatively free of compatibility issues and that procedures used with Scal in CSE116 will carry forward fine.

1.1 Student Learning Outcomes

Upon completing this course, you will be able to... (per CS/CE categories)

1. Compute, compare, and analyze runtime and function growth using asymptotic notation. (1,2,5,6/1,2,5,7)
2. Identify functionality of basic data structures. (1,2,6)/(2,7)
3. Identify the tradeoffs of different data structures, given their implementation. This also includes recognizing which situations benefit or suit the use of one data structure over another. (1,2,5,6/1,2,5,7)
4. Use data structures in programming. (1,2,5,6/1,5)
5. Implement and analyze basic algorithms such as searching and sorting, as well as recursive algorithms, tree traversal algorithms, and graph traversal algorithms. (1,2,5,6/1,2,5,7)

References:

- ABET: <http://www.abet.org/>
- CS: <https://engineering.buffalo.edu/computer-science-engineering/undergraduate/degrees-and-programs/bs-in-computer-science/accreditation-assessment.html>
- CE: <https://engineering.buffalo.edu/computer-science-engineering/undergraduate/degrees-and-programs/bs-in-computer-engineering/accreditation-assessment.html>

All will be touched by exams and written assignments, and programming itself will touch all but 1 and 3. As judged by the ABET guidelines, the CS outcomes 1,2,5,6 are supported, as are the CE outcomes 1 and 5, while CE 2 and 7 are minimally supported.

1.2 Academic Honesty

A university is a *community*, and every community has values and rules that go hand-in-hand with membership in the community. At universities one rule is the standard of *academic honesty* as it has been understood and followed for **all** of the just-ending millennium. This rule is not written down in a standard text such as Magna Carta or the Constitution, but is the same for every educational institution even though they all have individual statements of it. The CSE Department now requires that students in every course have read UB's statements of the rules, which are now online and collected as links on the page <http://www.cse.buffalo.edu/shared/policies/academic.php>.

In this course, almost all assignments will be individual; the lone exception being paired teams on the final project (mainly on its second stage). For those, the rule takes a particularly simple form: All assignments must be *your own work*. This term does not need a definition or legal parsing. We have a whole course, CSE442, to teach teamwork to mature programmers, but 115–116 and 250 are *formative* courses. Even those communities that gave us the maxim “It Takes a Village” have *solo* formative experiences, such as a “walkabout.” Even sitting at adjacent terminals in an on-line cafe and sharing routines over a cup of Java is strictly *ixnay*—such things invariably *eScalate* out of control.

Of course, this policy has to be reconciled with students' desire for help on assignments. The Department has a reasonable guideline tailored to programming projects at the above URL. We will talk about "reasonable discussion" of problems in class at some convenient and good time.

On *weekly written homeworks*, in the ancient times when those were collected in hardcopy, the largest kind of cases were [*claimed to be*] one student copying off another who couldn't attend class and asked the "trusted friend" to submit for him/her. In such cases *both* students receive a zero on the set with a notation made in their files; and it is only for "first offenses" of this nature that we tend to show this kind of leniency. Nowadays, however, the biggest scourge is students taking photos of completed assignments and texting those photos to others. This shows *premeditation* to a degree that goes with a *second offense* (from another course), and cheating on a longer *project*, all liable for an automatic F in the course and/or expulsion from the Department (*cum*-University).

One other principle is that *use of local resources still needs to be cited*. If you are adding code to a file already partly written, then chances are the header comment will already amount to such a citation—though it is still a good idea to mark where your adds or changes begin with `/**` style comments. When you are transferring code from a separate source, even one of

- the textbook,
- sample code on our webpages,
- any code from the official Scala websites, or
- any code otherwise linked from course websites or Piazza responses, or
- code otherwise provided by instructors/TAs,

that use *still needs to be cited*. And of course, **one needs to cite any outside sources used**.

More specific information will be given out on assignments, readings, and individual/joint-work policies.

1.3 Code Examples and Projects

Many code examples (some downloaded via WWW) are available now in my `~regan/cse250/...` directories on the CS machines. This course is intended to be a good experience (and possibly a first experience) with software systems of appreciable size. The philosophy is basically: rather than ask you to write 500 lines from scratch, give you an 800-line system and ask you to add 200-500 lines to it. The system for use this term is being built now in consultation with TAs. Further code examples will be added during the term. If you find *more* good stuff on the WWW, please let us know! We are trying hard to devise reasonable projects whose answers can't be downloaded. (However, don't fool yourself about this—we can often recognize code that you didn't write.)

Our policy is that *late work is not acceptable*. In return, you get an answer key shortly afterward, and a quick turnaround of graded work before the next problem set is due.

The remaining material mostly quotes information from the Fall 2021 CSE250 policies posted at <https://odin.cse.buffalo.edu/teaching/cse-250/2021fa/index.html>.

1.4 Incompletes

University policy specifies that students may be given an “I” grade only if they have a passing average in coursework that has been completed and have well-defined parameters to complete the course requirements that could result in a grade better than the default grade. It is intended for cases such as an auto accident or similar unforeseen circumstance. It is not intended as a “do-over” for those falling behind; the action specified by the University for such cases is resignation/withdrawal, for which the deadline this term is **Friday, April 22**. Contingencies with financial aid and other status matters are your responsibilities and not instructional concerns of this course.

1.5 Accessibility Resources

If you have any condition that requires reasonable accommodations to enable you to participate in this course, please contact the Office of Accessibility Resources in 60 Capen Hall, 716-645-2608; <http://www.buffalo.edu/studentlife/who-we-are/departments/accessibility.html> My first-day survey has an entry saying also to e-mail me separately in such cases. If you are in genuine doubt after having not said anything in that item, let me still encourage you to contact them—I’ve found the working relationships to be excellent. That continues into the next item.

1.6 Mental Health Services

As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, Covid, anxiety, Covid anxiety, high levels of stress, alcohol/drug problems, feeling down, feeling down about Covid, health concerns, or unwanted sexual experiences. UB Counseling, Health Services, and Health Promotion are there to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

- Counseling Services: <https://www.buffalo.edu/studentlife/who-we-are/departments/counseling.html> 120 Richmond Quad (North Campus), 716-645-2720, and 202 Michael Hall (South Campus), 716-829-5800.
- Health Services: Michael Hall (South Campus), 716-829-3316
- Health Promotion: 114 Student Union (North Campus), 716-645-2837