CSCE 322 Syllabus

Vital Information

Course Number	CSCE 322
Section Number	001
Course Title	Programming Language Concepts
Credits	3
Days	Monday, Wednesday, & Friday
Time	10:30am - 11:20am
Location	AVH Room 106

Prerequisite Courses

- CSCE 156/H or RAIK 184H or CSCE 311 or SOFT 161/H

Textbooks

Concepts of Programming Languages (ISBN-13: 978-0-13-139531-2)

Instructor Information

Lecture

Name	Ryan Patrick
Office	Avery 357
Phone Number	402-472-5089
Email Address	rypat@unl.edu
Website	http://cse.unl.edu/~rpatrick
Office Hours	TBD
Office Hours Location	TBD

About This Course

From the Undergraduate Bulletin

List-processing, string-processing, and other types of high-level programming languages. Fundamental concepts of data types, control structures, operations, and programming environments of various programming languages. Analysis, formal specification, and comparison of language features.

My Learning Objectives for CSCE 322

Upon earning a C, P, or better in CSCE 322, a student will be able to

- Implement solutions to novel problems in languages that go beyond the imperative and object-oriented paradigm.
- Demonstrate an understanding of feature of different languages (such as variations in scoping and binding and control flow).

- Demonstrate an understanding of how solving a problem in one paradigm can inform solving the same problem in a different paradigm.
- Apply an understanding of how programs in different paradigms would produce the same solution to the same problem.

Course Policies

Face Coverings

Given current CDC guidance and the current transmission level of COVID-19 in our community, I respectfully request that you join me in wearing a face covering during our classes.

Ordering of Topics

Week(s)	Topic(s)	Concepts of Programming Languages	Grades
1	Introduction & Syntax	Chapters 1 & 2	
2	Syntax & Names	Chapters 3 & 4	
3	Semantics	Chapter 3	Homework 01 Out
4	Semantics & Control Flow	Chapters 3 & 8	
5	Exceptions	Chapter 14	Midterm 01
6	Scoping & Binding	Chapter 5	Homework 01 In, Homework 02 Out
7	Data Types & Data Abstraction/OO	Chapter 6	
8	Data Abstraction/OO & Functional Languages	Chapters 6 , 11, 12 & 15	Homework 02 In, Homework 03 Out
9	Functional Languages	Chapter 15	
10	Functional Languages	Chapter 15	Midterm 02
11	Functional & Logic Languages	Chapters 15 & 16	Homework 03 In, Homework 04 Out
12	Logic Languages	Chapter 16	
13	Logic Languages	Chapters 16	
14	Logic Languages & Scripting Languages	Chapters 13 & 16	Homework 04 In
15	Logic Languages, Scripting Languages, & Review	Chapters 13 & 16	
	Final Exam		Final Exam

Tentative Course Calendar

Sun	Mon	Tue	Wed	Thu	Fri	Sat
22	23 10:30am - Lecture AVH 106	24	25 10:30am - Lecture AVH 106	26	27 10:30am - Lecture AVH 106	28
29	30 10:30am - Lecture AVH 106	31	1 10:30am - Lecture AVH 106	2	3 10:30am - Lecture AVH 106	4

CSCE 322, Holidays in United States

Sep 2021 (Central Time - Chicago)

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30 10:30am - Lecture AVH 106	31	1 10:30am - Lecture AVH 106	2	3 10:30am - Lecture AVH 106	4
5	6 Labor Day	7	8 10:30am - Lecture AVH 106	9	10 Homework 01 Out 10:30am - Lecture AVH 106	11
12	13 10:30am - Lecture AVH 106	14	15 First Day of Hispanic Heritage 10:30am - Lecture AVH 106	16	17 10:30am - Lecture AVH 106	18
19	20 Midterm Exam 01 Window 10:30am - Lecture AVH 106	21	22 10:30am - Lecture AVH 106	23	24 10:30am - Lecture AVH 106	25
26	27 10:30am - Lecture AVH 106	28	29 10:30am - Lecture AVH 106	30	1 Homework 01 In Homework 02 Out 10:30am - Lecture AVH 106	2

Sun	Mon	Tue	Wed	Thu	Fri	Sat
26	27 10:30am - Lecture AVH 106	28	29 10:30am - Lecture AVH 106	30	1 Homework 01 In Homework 02 Out 10:30am - Lecture AVH 106	2
3	4 10:30am - Lecture AVH 106	5	6 10:30am - Lecture AVH 106	7	8 10:30am - Lecture AVH 106	9
10	11 Columbus Day Indigenous Peoples' Day 10:30am - Lecture AVH 106	12	13 10:30am - Lecture AVH 106	14	15 Homework 02 In Homework 03 Out 10:30am - Lecture AVH 106	16
17	18	19	20 10:30am - Lecture AVH 106	21	22 10:30am - Lecture AVH 106	23
24	25 Midterm Exam 02 Window 10:30am - Lecture AVH 106	26	27 10:30am - Lecture AVH 106	28	29 10:30am - Lecture AVH 106	30
31 Halloween	1 First Day of American Indian 10:30am - Lecture AVH 106	2 Election Day	3 10:30am - Lecture AVH 106	4	5 Homework 03 In Homework 04 Out 10:30am - Lecture AVH 106	6

Sun	Mon	Tue	Wed	Thu	Fri	Sat
31 Halloween	1 First Day of American Indian 10:30am - Lecture AVH 106	2 Election Day	3 10:30am - Lecture AVH 106	4	5 Homework 03 In Homework 04 Out 10:30am - Lecture AVH 106	6
7 Daylight Saving Time ends	8 10:30am - Lecture AVH 106	9	10 10:30am - Lecture AVH 106	11 Veterans Day	12 10:30am - Lecture AVH 106	13
14	15 10:30am - Lecture AVH 106	16	17 10:30am - Lecture AVH 106	18	19 10:30am - Lecture AVH 106	20
21	22 10:30am - Lecture AVH 106	23	24	25 Thanksgiving Day	26 Native American Heritage Day	27
28	29 10:30am - Lecture AVH 106	30	1 10:30am - Lecture AVH 106	2	3 Homework 04 In 10:30am - Lecture AVH 106	4

Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29 10:30am - Lecture AVH 106	30	1 10:30am - Lecture AVH 106	2	3 Homework 04 In 10:30am - Lecture AVH 106	4
5	6 10:30am - Lecture AVH 106	7	8 10:30am - Lecture AVH 106	9	10 10:30am - Lecture AVH 106	11
12	13 Final Exam Window	14	15	16	17	18

Grading

A+ ≥ 97	B+ ≥ 87	C+ ≥ 77	D+ ≥ 67	F < 60
A ≥ 93	B ≥ 83	C ≥ 73	D ≥ 63	
A- ≥ 90	B- ≥ 80	C- ≥ 70	D- ≥ 60	

Grades of Incomplete will only be given in accordance with University policies.

Weighting

Assignment	Weight
4 Assignments	$3 \times 15\% + 1 \times 10\%$
2 Midterm Exams	$2 \times 10\%$
Final Exam	15%
Clicker Questions	5%
Preprocessing Questions	5%
Total	100%

Academic Integrity

Academic Integrity Policy (Department of Computer Science & Engineering)
While I encourage discussion, I abhor academic dishonesty. Discussion should be done at a level above pseudocode. Discussions of *what* and *why* are perfectly acceptable, but discussions of *how* something is implemented are prohibited.
According to the Department's policy

> Unless specifically prohibited by the instructor, it is acceptable to discuss the meaning of assignments. Discussing general approaches and strategies for solutions may be permissible, but unless specifically allowed, such communications should not include written material or code and should not transmit substantive or specific elements of a solution.

Examples of permitted diagrams to illustrate programming concepts can be found in figures 15.1, 15.2, and 16.1 of the textbook.

The use of solutions provided by other students (both current and former), "solution manuals", and other external sources is prohibited.

`handin` and `webgrader`

The webgrader is provided so that you can test how **your** submission(s) will behave when presented with data of the form that will be used to grade the submission(s). Because you must submit something before you are allowed to run the webgrader, it is not possible to "just run the webgrader on it". **You are responsible for everything submitted on handin under your CSE username**.
According to the Department's policy, "Students found guilty of academic dishonesty will be punished as outlined in the policy, which may include failing the course or expulsion from the University."

First-time offenders will receive a 0 on the entire assignment in question and be given an Incomplete in the class until they successfully complete an online ethics course administered by the Office of the Dean of Students. Repeat offenders will receive an F in the class and will be recommended to the Academic Integrity Committee for dismissal from the Department and all of its programs.

Course Policies

webgrader

The webgrader will be used in this class and, with very few exceptions, your output must match the solution output exactly to receive credit.

Late Work

Each student is given a total budget of 7 **calendar** days that they *can* spend on delaying the submission of assignments without penalty. Email me when turning an assignment in late (as I am not automatically notified). After this budget is exhausted, an assignment that uses an eighth day late will receive 50% credit. Assignments that are submitted after the eighth late day is used will receive no credit.

Pre-approved delays (by contacting me **beforehand**) and catastrophic delays (with documentation) do not affect the budget of 7 days.

The Department's Handin System considers 11 : 59 **to be** 11 : 59 : 00**, not** 11 : 59 : 59**.**

Clicker Questions

While covering each topic, I will ask a number of clicker questions in class. These multiple choice questions are designed to measure your understanding of a topic that is being covered and challenge you to think about how the concept may be used in the future.

"Preprocessing" Questions

Throughout the semester, I will periodically ask questions that require an open-ended, online response. These questions are asked to introduce you to a topic that we will be covering in class and to gauge the familiarity that the class has with the topic.

Questions about Grading

Questions about grading must be submitted via email and are only accepted in the 7 calendar days following the posting of a grade.

Course Announcements

Announcements will be made through Canvas or email. The Department expects students to regularly check their email, so that they do not miss important announcements.

Anonymous Suggestion Box

The Department of Computer Science & Engineering maintains an anonymous suggestion box that you may use to voice your concerns about any problems in the course or department if you do not wish to be identified.

Student Resource Center

The Department maintains a facility where students can go for additional assistance with coursework. It is located in Avery Hall 12 (formerly Avery 13*A*).

Services Provided by UNL

UNL offers a variety of options to students to aid them in dealing with stress and adversity. Counseling and Psychological Services (CAPS) is a multidisciplinary team of psychologists and counselors that works collaboratively with Nebraska students to help them explore their feelings and thoughts and learn helpful ways to improve their mental, psychological and emotional well-being when issues arise. CAPS can be reached by calling 402-472-7450. Big Red Resilience & Well-Being provides fun events, innovative education, and dynamic services to help students understand emotions, manage stress, build strength, connect with others, develop grit and navigate transitions.

Students with Disabilities

Students with disabilities are encouraged to contact me (the instructor or teaching assistant) for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Nebraska-Lincoln to provide flexible and individualized accommodations to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, students must be registered with the Services for Students with Disabilities (SSD) office, 132 Canfield Administration, 472-3787 voice or TTY.

Tips & Tricks

Teaching Philosophy

At its core, the field of computing is about divising approaches to problems that have never been solved before. Even on a small scale, a form collecting data to be inserted into a database for instance, if someone had previously done *exactly* what you are trying to do on *exactly* your system, then nobody would need you; they would just use the other implementation.

Preparing students for a career of always needing to make something new because the old stuff doesn't *quite* meet the requirements anymore necessitates an emphasis on thinking about problems from different perspectives and always asking, "Why?" To that end, I start the first class of every course, every semester with a seemingly simple clicker question: True or False, July and August are the only two consecutive months that each have 31 days? I want a question that everybody has the prerequisite knowledge to answer, but the answer is not intuitive. I have the students answer first on their own (usually no more than 60% of students can agree on an answer), then I show them the polling results and ask them to convince their neighbor that they have the right answer. After that, I poll the class again (and the results are usually more decisive). As developed, Eric Mazur's method of peer instruction doesn't intend for the instructor to reveal the poll results when students first answer individually, but I do it to make a point: a seemingly "easy" question with an "obvious" answer isn't always so easy and obvious. When I asked the question during a sample lesson in an education class I was taking, one of the grad students turned to the class and exclaimed, "Do you not know the knuckle rule?!" After a moment, the student realized the mistake they had made. Once I can establish to the class that "easy" questions might not have such obvious answers, they take future questions under more serious consideration.

To get students to always ask, "Why?" I provide randomly-generated, constantly-updating test cases for programming assignments (and access to a system that will allow students to run their code on these test cases and corresponding solutions). While students' final grades are determined by performance against a static set of test cases that everyone's submission is run on, the test cases constantly change up until an assignment is due. The purpose is simple: to help students find faults in their approach. In upper-level classes, approaches can be complex. I can't account for all of the ways someone may slip up and handmake test cases that expose those faults. So, instead, I provide test cases that adhere to the description of what input will look like and leave everything else up to a random number generator. While a submission that gets every test case right or every test case wrong is uninteresting, my hope is that when a student sees some successes and some failures, the student will ask, "Why?" Programs are deterministic things; when something "unexpected" happens, the program didn't make the mistake, there was a flaw in the design. With random test cases, I hope to have a higher likelihood of revealing to a student something that either wasn't considered or wasn't correctly implemented.

Training future computing professionals is not about taking what is currently known about computing and trying to store as much in students' long-term memory as we can in four years. Teaching computing students is about training them to adapt to new problems that they will spend their entire professional career solving. Graduation is not the light at the end of a tunnel, it is the light at the end of a cannon barrel. Once students graduate, they are on a trajectory that will carry them for the duration of their professional lives.

Instructional Philosophy

I view instruction in three (3) parts. During the week, we only have 150 minutes (or less than 1.5% of a possible $10,080$ minutes in a week) to cover material in class. In that time, I want students to get an intuitive feel for the topic being covered. To do that, I will use more visual and interactive techniques, and leave more mundane tasks to outside of class.

Away from class and office hours, the textbook(s) (and other, written materials) become almost the sole source of information. Because of this, I prefer a textbook that contains a lot of detail, over a textbook that only provides brief explanations. Outside of class and office hours, your classmates will also become a source of information. Unlike the physical sciences, computer science does not have examples that can be observed in nature. To completely understand something that cannot be sensed, multiple interpretations of the subject might be necessary. This is where discussions with classmates come in. While I may be able to provide multiple analogies to communicate a topic, my analogies may not be enough. Discussing *topics* with classmates may provide you with an understanding of a concept by providing an analogy that is relevant to *you*.

Between time spent in class and time spent reading the textbook lies office hours. The primary purpose of my office hours is to directly connect the intuitive understanding that was obtained in class to the abundance of information that the textbook and other materials provide. Office hours are not a substitute for class or work/discussions away from class, they are a time to draw connections between time in class and time away from class.

Ask a question . . .

in class if you don't get an intuitive feel for a topic that is being covered While I try to provide analogies and activities that are relevant, an analogy that is relevant to 80% of students is still irrelevant to 8 students in a class of 40. Being in the group that didn't find an analogy relevant is like being in a group that didn't find a joke funny: sometimes, it's the messenger's fault. If you have a question about something in class, you're likely not alone. I will try to come up with an analogy that is relevant.

anytime if the textbook and materials aren't sufficient If there was a source that covered every topic in a way that every reader fully understood, writing another word on the subject would be redundant. A quick search will show you that classes like this are being taught all over the world and they're not all using this textbook. A textbook that perfectly covers 9 out of 10 topics still has its flaws that another source may fix.

in office hours if you don't see the connection between class and the textbook Once again, this might be the messenger's fault and you are likely not alone. We all come to this class with different experiences that lead us to interpret things differently. What I may think is a good connection between class material and the textbook may not be effective the first time.

Campus Resources

Online Resources for Undergraduate Students (Department of Computer Science & Engineering)

Avery Hall

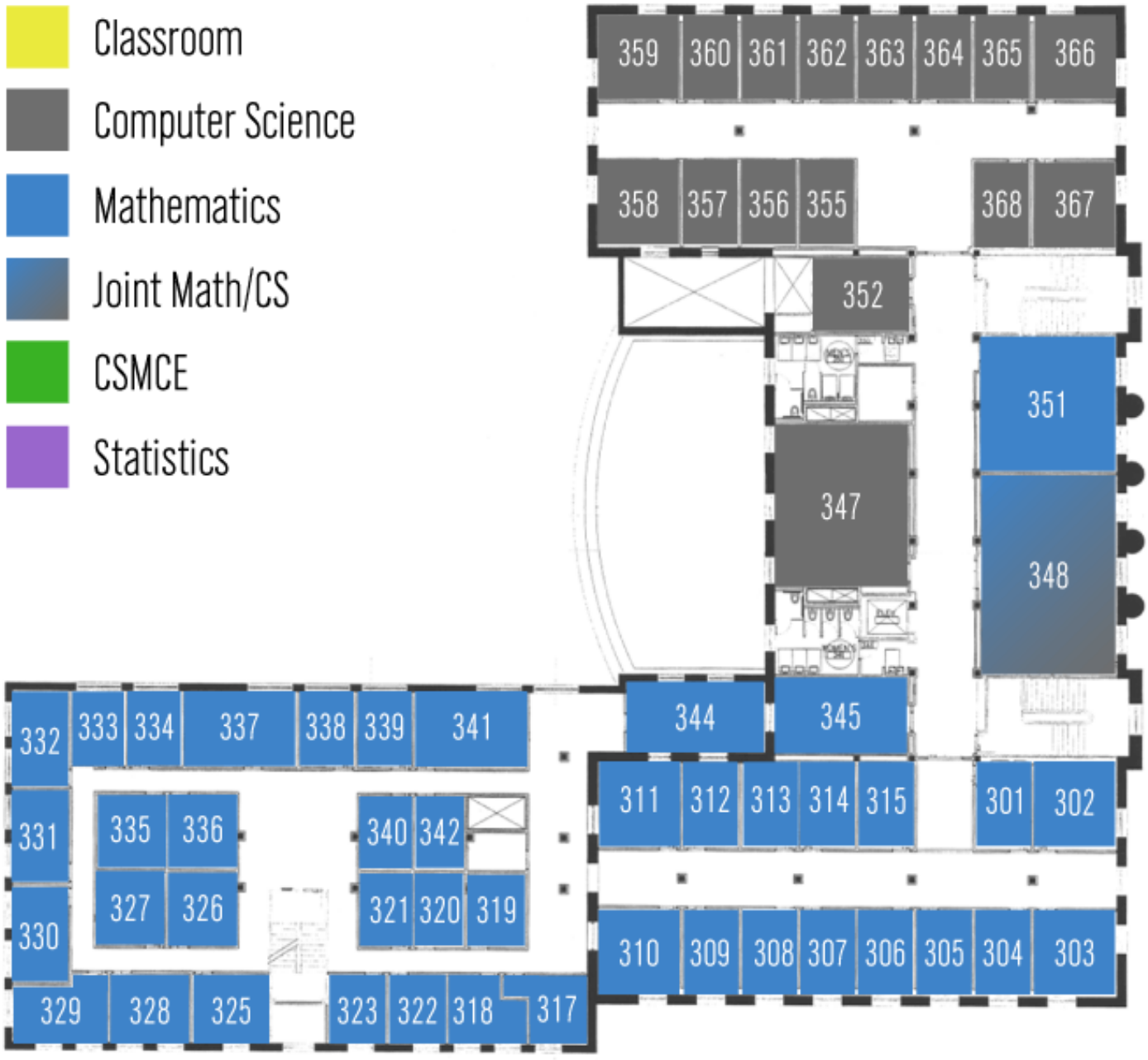

Figure 1: Floorplan of 3^{rd} Floor of Avery Hall

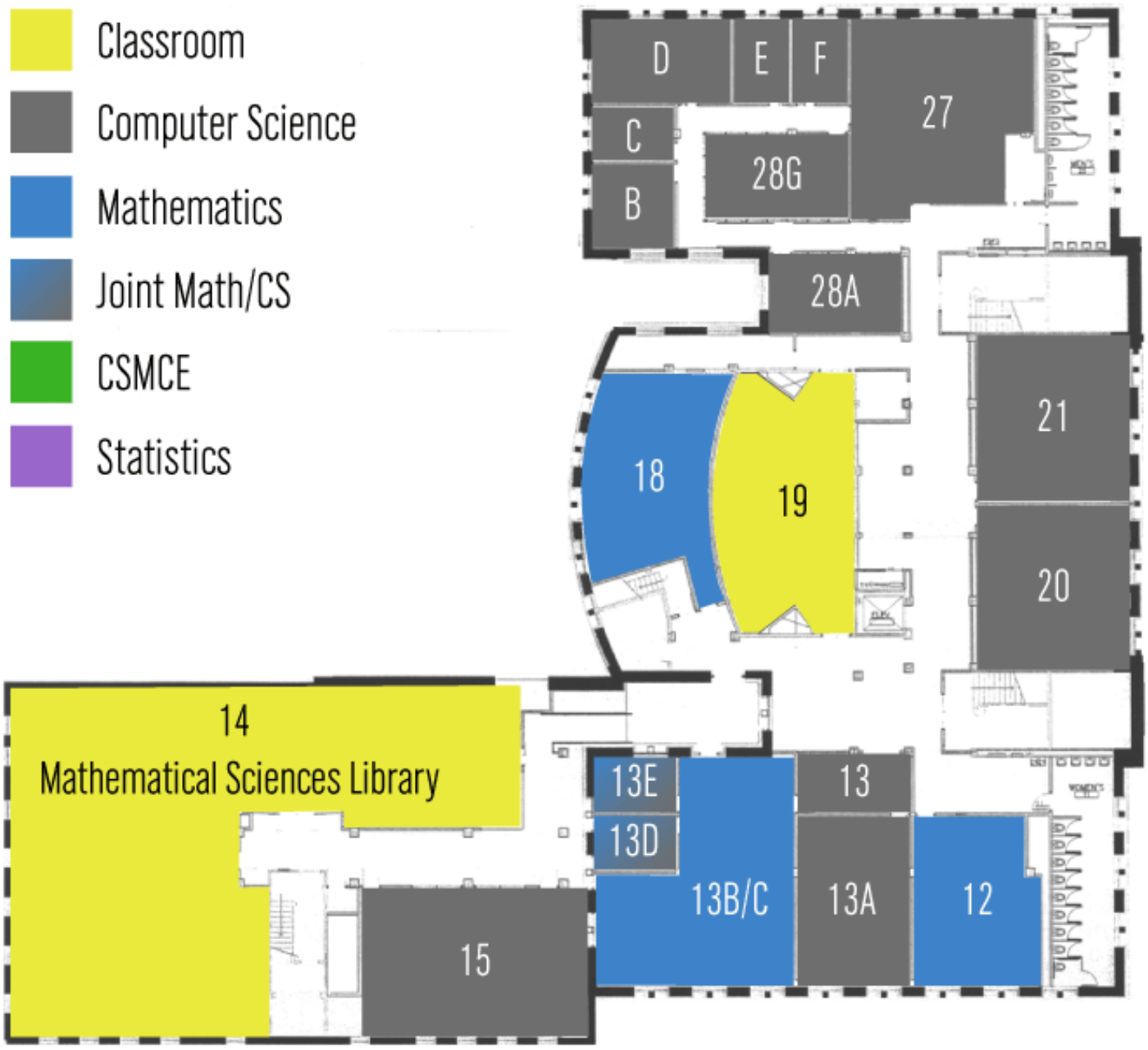

Figure 2: Floorplan of Basement of Avery Hall

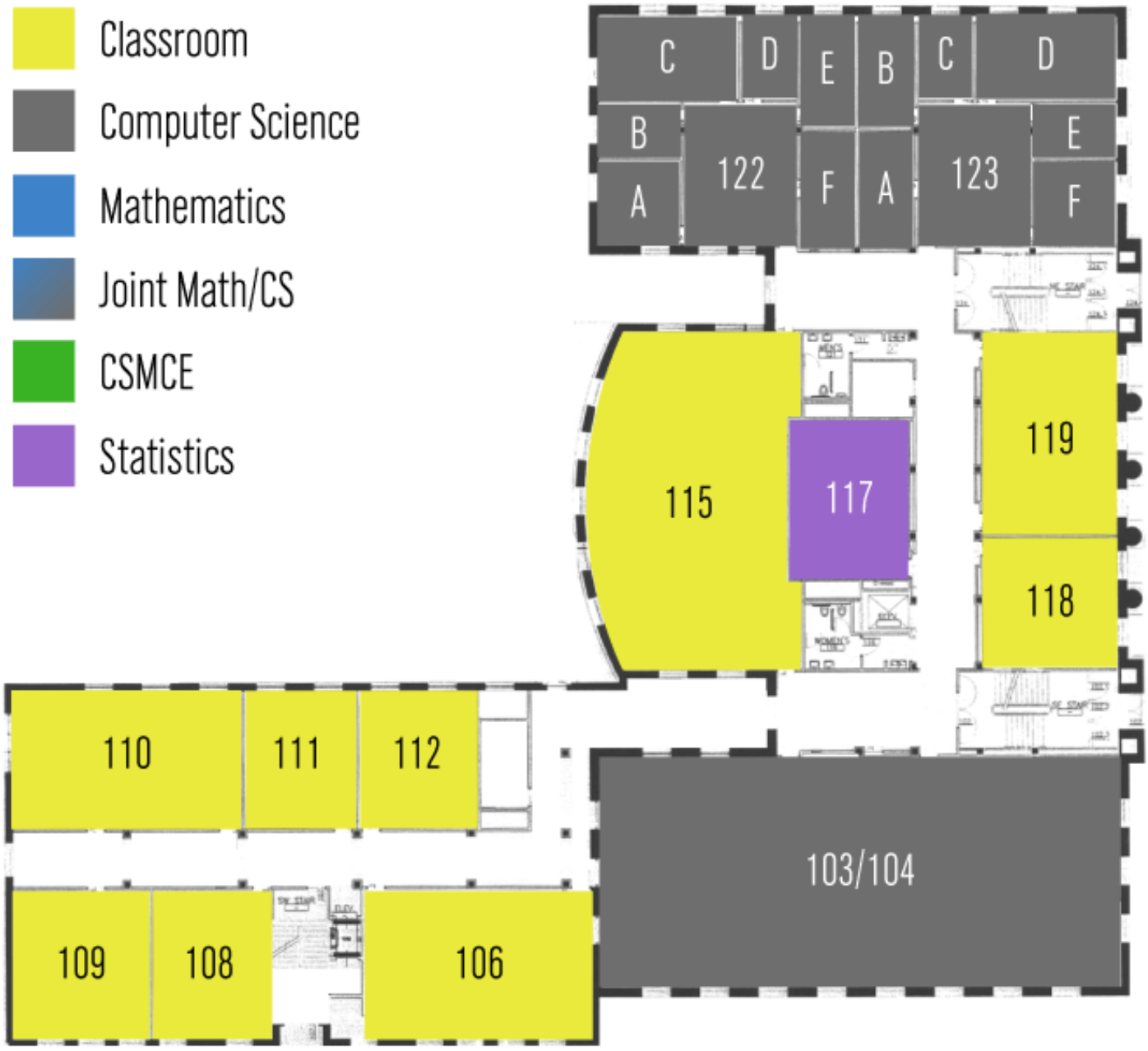

Figure 3: Floorplan of 1^{st} Floor of Avery Hall