Syllabus: CIS 300 Foundations of Computer Science

Brian C. Ladd

Spring

Introduction

Catalog Description

An introduction to the logical and quantitative foundations of computer science. Topics include introductions to: formal proof techniques, logic, sets, relations, partial order, number systems, combinatorics, graphs and trees, and matrix arithmetic. Prerequisites: CIS 201. Spring.

Overview

This course is the gateway to theoretical computer science. Discrete mathematics is the study of discrete, or separate, objects. Where calculus studies continuous functions (and hence real numbers), discrete mathematics studies separate objects, functions on them, sets of them (and hence focuses on integers).

This course begins by laying out first-order predicate logic and describing how to use axioms, logic, and inference rules to prove mathematical theorems. Discrete structures such as sets, matrices, functions, and graphs are then presented and used as fertile fields to practice proving theorems.

In the middle of this course there is a section on mathematical induction and recursion. This is, perhaps, the most important part of the course, as it is proof by induction that makes possible many of the proofs underlying the theory of computation (and certainly complexity).

Learning Objectives and Outcomes

The Computer Science program of the CSOLT department has adopted ten learning objectives for students completing an undergraduate major or minor in the department. Each course in the curriculum is designed to address particular objectives so that the overall curriculum fulfills our goal of producing capable computer scientists.

1Yes, it is only a theorem. Of course in mathematics, being only a theorem means that it has been rigorously proved to be true.
The objectives (and broad outcomes) serve as a rubric for evaluating the learning that happens in this class. You may wish to review this section of the syllabus when filling out the course evaluation and especially when studying for midterm and final exams.

**Objectives**

Students in the Computer Science program at SUNY Potsdam are expected to graduate with a:

1. knowledge of discrete and continuous mathematics – including elementary probability and statistics – and the ability to apply logic and mathematical proof techniques to computing problems.
   
   *Students understand how to translate natural language statements into quantified and non-quantified statements in first-order predicate logic. Students can invert statements in first-order predicate logic. Students can write truth tables.*
   
   *Students can apply rules of inference, truth tables, and weak and strong induction to prove mathematical theorems.*

2. knowledge of basic theory of computability and complexity of computation.
   
   *Not applicable to CIS 300.*

3. knowledge of and the ability to apply programming fundamentals in at least two programming languages.
   
   *Not applicable to CIS 300.*

4. knowledge of fundamental data structures and algorithms – including analysis of their correctness and complexity – related to various fields of computer science, and the ability to apply this knowledge to problems through the use of appropriate programming languages.
   
   *Students are expected to be able to describe one or more computer representations of discrete data structures: sets, matrices, and functions. They are also expected to be able to discuss running-time implications of different representations.*

5. knowledge of computer architecture and organization, computer operating systems, and computer networks, and the ability to apply this knowledge to problems through the use of appropriate programming languages.
   
   *Not applicable to CIS 300.*

6. competence and effectiveness in technical oral, written, and visual communication, particularly as they apply to the dissemination of technical information on subjects dealing with computing technology and applications.
   
   *Not applicable to CIS 300.*

7. knowledge of and skill in applying good practices in software engineering.
   
   *Not applicable to CIS 300.*
8. the ability to function effectively in teams to accomplish a common goal.

_Not applicable to CIS 300._

9. an understanding of professional, ethical, legal, security, and social responsibilities and issues, including an awareness of impact of computing on individuals, organizations and society.

_Not applicable to CIS 300._

10. a commitment to continuing professional development.

_Not applicable to CIS 300._

**Outcomes**

Upon completing this course, students should be able to

- Prove discrete mathematics facts using direct, indirect, and inductive techniques.
- Explain the relationship between induction and recursion.
- Define and operate on sets with union, intersection, compliment, and cross product. Apply the powerset operator (and count the members in the result).
- Define and operate on relations: transitive closure, determine predicates that apply (antisymmetric, symmetric, transitive, associative, commutative).
- Understand the definition of equivalence relations. Change between partition and equivalence relation notations.
- Be able to generate permutations and combinations of objects from sets.

**Grading**

This is a 300-level course in one of your major fields. You will be evaluated on both your thinking about the course content and your ability to communicate your thinking. This means that your writing must be clear, concise, and in proper English (the language of the course). The proofs should be complete with a statement of givens, that which is to be proven, and a logical progression of statements proving it.

- **In-class Participation** 10%
- **Weekly Quizzes** 35%
- **Tests** 30%
- **Final Exam** 25%

**In-class Participation** This semester we are trying something new: a **flipped classroom.** That means class time will be spent working lots of sample problems in groups of size two and three. Your participation with your peers is expected every class.
Weekly Quizzes: Wednesday quizzes on the previous week’s material. These quizzes will be strictly time limited. The goal is for each quiz to count for about 5% of your grade; you do not want to miss quizzes as there will be no make-ups.

Tests: In-class exams primarily covering material that is new from the last test (I reserve the right to re-test material if I deem there is a need for that).

Final Exam: Cumulative written final exam; questions drawn from earlier exams, reading, lectures, and homework.

Textbook and Readings

General Rules

Your choices make your fate. Information arms you to make your best decisions and enjoy the best fate. This section describes the class expectations so that you can meet them (or decide not to).

Instructor Expectations of Students

Communication

Read/respond to e-mail. Read/respond to the course Moodle site. These are the two primary means of communication in the class. You should make use of them.

Moodle is the communications hub of this class. It is where readings, writings, programs, presentations, and homework deadlines appear. If you find any discrepancies or have questions about due dates or the meaning of assignments ask in person, in class, or in e-mail. My office hours are listed in my Moodle profile.

Students are expected to have a copy of the textbooks (if any) and are expected to complete reading assignments before the beginning of class the day they are due. In-class participation requires you to have engaged the reading.

Students are expected to listen actively in class. They are also expected to take notes in class. These two activities (which are linked) correlate strongly with understanding the material presented. Do not copy slides or board notes directly: digest them and restate them in your own words. Mark where you expect test questions to lurk (sometimes the instructor will even help with this).

Advanced courses often use a version control system (such as git) to retrieve and submit source code and programming assignments. This requires you to understand how to use git, to understand the difference between the class repository (which belongs to the instructor) and assignment repositories (which belong to you).

Attendance

American undergraduates are old enough to join the army, vote, and even get married. You are each mature enough to make your own decisions about attending class. Be aware that you decisions have consequences.
Students are expected to attend every class. Students are responsible for all material covered in every class meeting. There is no taking of attendance in this class (see 'old enough' in the previous paragraph).

The study of computer science is cumulative; past experience shows a strong correlation between high absences and low grades.

The registrar schedules the final exams every semester; I am not able to move them and do not give early finals. Take-home final exams are due before the end of the scheduled exam period (it will be accepted early).

**Do Your Own Work**

Do your own work. That should go without saying but it appears that for some students it bears repeating.

Code you plan to turn in for an individual grade should not be shared with any other student; you should never look at another student's project code, not even for 'guidance.'

I want you to discuss assignments and show off results of your playing with programming different things. Discussion at a high-level, even if it includes details on data structures is fine; debugging assistance is also acceptable though the depth of code analysis necessary for debugging skates on thinner ice; directing someone else on exactly what methods a given class should have or the exact format of a programmer-controlled file should take is beyond the pale.

This does not apply to group work: in a group project, all members of the team are expected to see and understand all parts of the project. Cross-disciplinary understanding is a highly valued skill in the real world and a major reason for group work in our curriculum.

**Turning In Work**

Start early, everything takes longer than you think.

As mentioned earlier, reading is to be done before class. Homework assignments are also due at the beginning of class so please complete them before class.

Assignments have a due time: a date and a time when they are due. If they are due electronically (they all are), make sure you submit them the expected way by the expected hour.

Make sure you check the spelling and arithmetic in your assignments. It does not help your grade (or my mood) to turn in sloppy work.

**Attitude**

Learning is not always easy. Learning is not always comfortable. Students must actively engage the material and be able to ask for help. This does not mean look at the problem for fifteen seconds and throw up your hands because it is 'hard.' It does mean beating your head against it for a bit and then asking me for guidance.

Be respectful. Trust me that I actually have a plan and I know where we are going (okay, I don’t always know exactly where we are going at every moment but I do have a plan). The readings, the assignments, the lectures, they all go together.
Start early! Everything takes longer than you think.

Student Expectations of the Instructor

Communication

You may expect me to state and keep office hours (you know where my office is, right? 301 Dunn Hall) They are part of my Moodle profile.

You may expect fairly prompt answers to e-mail (next morning is typical).

You may expect confidentiality: I will not discuss your grades with anyone except my departmental colleagues (and then only to make sure that I am doing my job correctly). I will not share work you have done without your permission.

Attendance

You may expect me to be prepared. This means for class every day and it means with assignments, too. You should expect adequate time to complete an assignment after it is published. You may expect some flexibility (with in the limitations of getting through the course materials) in due dates.

Assignments

Expect assignments to be clear (correct spelling, correct grammar). Expect most sample code to compile and run. Expect assignments to be accompanied by a clear system by which they will be evaluated. Expect me to follow that system.

Students should expect that their work will be evaluated on its own contained merits and not based on how closely it follows the instructor’s personal/political views (unless I put parroting my political views into the rubric).

Student Expectations of One Another

Wait, students have expectations of one another? Yes, they should. The most important expectation we should all have of each other is to create a safe learning environment. By safe I mean we should all feel safe to ask questions, to admit mistakes, to try and to be wrong.

Students should be willing to help one another and should all be aware of the limits.

In group work, all students in the group should expect equitable division of labor and should expect to evaluate every member of the group.

Students should expect to evaluate other students’ work (and to be evaluated by other students). Both parts of this process have valuable learning outcomes.

Students should expect to feel safe being different. What does this mean in computer science? I actually am not sure. It means respecting each other, evaluating each other in openly communicated terms related to computer science (since that is the topic of the course), and being flexible, accommodating each other whenever possible.
Grading Grievances

When graded work is returned you may have questions about how it was graded. If the question pertains to process (e.g., addition, number of points per subsection), by all means ask me as soon as possible. Typically, if the grade is changed, you will be asked to e-mail me the grade update so that the Moodle grade book is current.

If, however, you feel that your grade fails to reflect the quality of your work (you want to argue that your answer is, in fact, correct, rather than incorrect), take a deep breath and wait. You must wait no less than one and no more than seven days after the work is returned to you to raise the issue with me.

After the cooling-off period, resubmit the work along with a written explanation of your concern. This is to assure that your concern gets a cool, thoughtful consideration. If I agree to regrade the work, I will regrade the entire project, replacing the old grade with the new one.