Course Description

This is a required course for computer science majors that is to be taken at the end of the sophomore year or the beginning of the junior year. The course will use a functional language to introduce a high-level approach to programming over complex data. It will emphasize a view of such data that abstracts away from their representation, using types as a vehicle for organizing them as values and for structuring computations over them. Advanced programming techniques that use ideas such as recursion, higher-order functions, lazy and eager forms of evaluation and infinite data objects will be explored. The possibility of exploiting parallelism arising from pure forms of expression evaluation will be examined. Other techniques and principles to be studied include search-based programming, modularity and concurrency. Programming projects that focus on symbolic computation will be used in a central way to impart the core ideas in the course; such projects may include writing parsers, type-checkers and interpreters for suitably circumscribed programming languages, and applications of search-based techniques.

Course Topics

The topics to be covered are described below. This is not intended as a week-by-week schedule: material under different topics will be interleaved and reordered in an actual plan for the course.

- Types as an organizing principle for programming. Types as a language, higher-order and polymorphic types, types as means of classifying values, ad hoc & parametric polymorphism.
- Expressions and computation as effect-free evaluation. Binding of names, scoping, environments, closures; strict and non-strict evaluation, opportunities to exploit parallelism; lazy evaluation as a programming technique, infinite data structures; recursive functions and relation to recursive data; iteration as tail recursion, translating general recursion to tail recursion.
- Recursion and relation to inductive reasoning, invariants over functions, types and invariants, designing functions around invariants.
- Functions as first class objects, higher order functions (map, filter, fold) and applications, parametric polymorphism, functions as parameters, continuation passing style.
- Effects and computation. Type safe references, assignments, other side-effecting constructs, iterative control structures; modeling effectful computation via state transforming functions, effects in lazy languages (monads); object oriented programming as combining environments with state; references and circular data structures.
- Programs and analysis of complexity. Recursive functions and recurrence relations; functional data structures, efficiency and programming techniques; mutable data and efficiency.
- Value-based programming and realization. Mapping data objects to memory; memory usage, copying versus pointing; garbage creation and automatic collection, memory management.
Search-based computation. Search as a computational paradigm and its applications; programming techniques for realizing search.

Role of modularity in programming-in-the-large. Interface specifications, abstract data types; language support for modular programming, interface checking as type checking; module composition as function application.

Concurrency. Asynchronous computation as a paradigm, coordination through communication; language mechanisms for organizing and controlling communication.

Translation of principles into programming in mainstream, non-functional languages.

Contact information, office hours

- **Instructor:** Eric Van Wyk, email: evw@umn.edu, phone: 612-625-0329.  
  Office hours: Monday, Wednesday, Friday from 2:30pm to 3:25pm, or by appointment; in Keller Hall 6-203.

- **Graduate Teaching Assistants:**
  - Sam Marquart, email: marqu317@umn.edu,
  - Ancy Sarah Tom, email: tommx030@umn.edu,
  - Parag Panda, email: panda025@umn.edu

- **Undergraduate Teaching Assistants:** There are several undergraduate teaching assistants who will be assisting in labs and holding office hours.

- Office hours for all TAs are listed on the course Moodle page, which is https://ay16.moodle.umn.edu/course/view.php?id=12397

- **Note:** Most questions should be posted on the appropriate Moodle forum. Otherwise email should be sent to the course email alias csci2041@cs.umn.edu.

Important Dates

There are two sections of CSci 2041, but they will be following the same material and the same schedule. All dates for mid-term exams and assignments will be the same for both.

- **Section 1** meets Monday, Wednesday, and Friday, 1:25pm - 2:15pm in Keller Hall 3-210.  
  The final exam is scheduled for Wednesday, May 10 from 1:30pm to 3:30pm.

- **Section 2** meets Monday, Wednesday, and Friday, 3:35pm - 4:25pm in Nicholson Hall 155.  
  The final exam is scheduled for Wednesday, May 10 from 8:00am to 10:00am.

There will be four quizzes during the semester. These are scheduled for February 10, March 3, March 31, and April 21. These dates all occur on a Friday.

Course Prerequisites

- CSci 1913 or 1933: Students need to have the degree of programming experience and maturity obtained from completing one of these courses.

- CSci 2011: 2041 builds on ideas from this course such as induction, recursion/recurrences, and logic.
Course Texts

There is no textbook for the class. However, we will make use of several online resources for the OCaml programming language and programming techniques covered in the class. These will be provided electronically from the course Moodle page.

Lecture format

Lectures are designed to be rather interactive and less like a traditional lecture. There will be exercises that we do in class. So come prepared to work. Most lectures will have a large “white board” component that will not be found in the slides. Lectures will present material not in the texts and not on the lecture slides, which will be available on the course web page in various formats.

Engagement

Learning is much more than simply acquiring a collection of facts; it is a process of assimilating knowledge, using it, applying it, and organizing it so that you understand the relationships between different concepts. It is an active process, not a passive one. Thus it is critical that you are actively engaged in the course. What does that mean?

- **Attending class:** My classes involve lots of discussion and in-class exercises. If you are not in lecture, then you miss out on this important way to learn the material. To be actively engaged in the course you will come prepared, having read the assigned reading and doing the exercises to prepare you for class. You will also be an active participant in classroom activities.

  I do not count attendance as part of your grade. It turns out that I do not need to. Those who attend and are engaged invariably do much better on assignments and exams that those that do not consistently attend class. The best way to do well in this course is to attend class.

- **Avoiding distraction:** Research has shown that people are very bad at multi-tasking. Your brain cannot do two things well at once and most of our attempts to do two things at once make us slower and, frankly, dumber.

  For this reason, I do not allow cell phones in class. These are to be stored in your bag or in your pocket and not held in your hand or laid on your desk.

  For similar reasons, I do not allow laptops in class. There may be a few occasions in which we want to use them as part of an in-class exercise, but otherwise they are to be put away. The distraction to you is significant but they also distract those around you. (If there is a reason that you feel you need to use your laptop to take notes, then please speak to me about it.)

- **You are responsible for routinely checking Moodle for updates and announcements.**

- **Electronic discussion forums will be available on the Moodle class site.** The TAs and I will be reading discussions on this and providing our input, but it will primarily be a forum for students to discuss course topics and questions (not solutions) about homework assignments. Asking and answering questions here is an important part of engaging in the class material.
Course work

You will complete several programming and “on paper” assignments to help you engage the questions laid out above and to learn the finer points of the material.

Several of these assignments will be programming assignments. Many of these will be programming assignments meant to give you experience with the different programming principles and techniques covered in the course. These will be turned in electronically on http://github.umn.edu.

Other assignments are written assignments, for example explaining how or why a certain program, perhaps yours, works or to argue or prove that a program meets some correctness specifications. These include using axiomatic semantics to construct proof of partial co Grading illegible homework can be exceedingly time consuming and thus we do not accept work that is not easy to read. It is suggested that you type your answers to the non-programming homework assignments, perhaps using \LaTeX, but if you choose not to, your answers must be written clearly and legible. Illegible answers will be marked as incorrect as we cannot grade what we cannot read.

Quizzes and exams are also an important part of learning and not used solely for evaluation. By properly preparing for an exam you have an opportunity to step back from the specific concepts of individual topics and see how the pieces fit together. The encourage you to do this, your are allowed a “cheat sheet” for the exams on which you can write whatever you like. This sheet must be an 81/2 × 11 inch piece of paper. You can use one side of it and it must be hand written. Organizing this document and writing it out by hand is a great way to learn the material. Many students report carefully creating their cheat sheet and then not using it in the exam because they’ve learned all the material they wrote on it.

We will also spend a considerable amount of time doing exercises in class. These are an important component of the course and thus it is important that you consistently attend lecture. We will collect some of these exercises to ascertain how well students understand the material and also, on occasion, to grade them. But their primary purpose is to help you learn the material in class, often by discussing your solution or questions your fellow classmates.

**Required Work**

The exams and homework assignments will draw questions from potentially all of the material in the specified sections of the assigned readings, even if this material is not covered in detail, or at all, in the lectures. Also, lectures may contain information not in the assigned readings, but you will be responsible for this information on the exams and homework as well. Thus, it is important that you attend lecture.

Numerous in-class exercises will be given during lecture. On occasion these will be collected and graded. But these will primarily be used as learning tools, not assessment tools.

Labs sessions consist of a small amount of work meant to be done in lab with an occasional bit of extra work to be done later. Sometimes this will be work that is meant to be done collaboratively and thus attendance is required.

Homework assignments, however, are to be done on your own. **Collaborative work is not allowed on homework assignments.** See the section on cheating below.

For non-programming assignments, it is suggested that you type your answers to the homework, but if you choose not to, your answers must be clearly legible. Grading homework is a time consuming process and illegible answers cannot be graded.
Grading

Your final cumulative score for the course will be determined based on your scores on homework assignments and exams as follows:

- Cumulative homework score: 40%
  - Homework assignments – 30%
    Different homework assignments will contribute different amounts to your cumulative homework score. This distribution will be determined as the course progresses.
    You are required to turn in all (outside-of-class) homework assignments in order to obtain a passing grade.
    Also note that your lowest single homework score (excluding any score lowered for cheating, see below) will be dropped in the calculation of your final grade.
  - Lab attendance and work – 8%
    Your lowest two lab work scores will be dropped in computing this score. You are required to attend all but two lab sessions of your choosing to pass the course.
  - Class participation, in-class exercises, contribution to forums – 2%

 To pass the class your cumulative homework score must be a passing grade, that is above 60%.

- Cumulative quiz and final exam scores: 60%
  - Each of the four quizzes is worth 10% of your final grade.
  - Final exam is worth 20%.

 To pass the class your cumulative quiz and exam score must be a passing grade, that is above 60%.

Your final letter grade will be determined by this final cumulative score. Typically, final grades are assigned on a scale not unlike the following: above 90% - A, above 80% - B, above 70% - C, above 60% - D, otherwise - F. Since each course instance of a course has different exams and assignments these numbers may vary from the scale above. You will be informed during the semester of tentative cutoff numbers.

Missed quizzes: There are no written make-up quizzes. However, if you notify the instructor prior to the quiz that you will miss it for a documented University-approved reason then an oral examination in Prof. Van Wyk’s office will be arranged as your make-up quiz. If you do not notify the instructor prior to the missed quiz then there will be no make-up.

Late assignments: are generally not accepted or allowed unless previous arrangements have been made with the instructor or are due to a documented University-approved reason.

You should be aware of the University Senate’s policy on make-up exams available at [http://www.policy.umn.edu/Policies/Education/Education/MAKEUPWORK.html](http://www.policy.umn.edu/Policies/Education/Education/MAKEUPWORK.html) and their policy on grading available at [http://policy.umn.edu/Policies/Education/Education/GRADINGTRANSCRIPTS.html](http://policy.umn.edu/Policies/Education/Education/GRADINGTRANSCRIPTS.html).

Be sure to keep regular track of your accumulating score to make certain that it is correct. You will be given regular feedback of your scores, either via Moodle or GitHub, and you should make sure that all your scores are correctly recorded. Check with the TA if you find errors. Errors must be reported no later than 2 weeks after the scores are posted.
Academic Integrity

You are encouraged to discuss homework problems with your fellow students. A large part of solving a problem is getting a precise and complete understanding of what the problem asks. This also helps to resolve any misunderstanding you may have of the problem or unintentional ambiguities in the problem description.

Discussing answers to problems, however, is not allowed. The work that you turn in to be graded is to be your own independent work representative of your independent thinking. Your discussions should stop long before you get to details of a solution. If you are still in need of assistance at this point, seek it from the TA or the instructor.

While the Internet is a wonderful resource for all kinds of information, you are expected to solve the homework problems on your own. Copying solutions from the Internet is cheating just as copying from fellow students is. Similarly, letting others copy your work is cheating and treated as such.

Discussing solutions to problems or copying solutions from others is considered cheating and there are penalties for such action.

Cheating does not help one learn the material and thus defeats the whole purpose of being in school in the first place. Also, the homework is intended as a warm up for the exams—if you don’t learn how to solve the problems by doing the homework then your grades on the exams will surely suffer.

Detection: Software is used to detect similar assignments. When flagged, these are investigated by hand. A determination that cheating has occurred when a *preponderance of the evidence* indicates as much.

Consequences:

- After the first offense of cheating one’s final grade is dropped on full letter grade. Thus, an A- turns into a B-, a C+ turns into a D+, etc.
- On the second offense, a grade of F is given for the course.

All incidents of cheating will be reported to the Director of Graduate or Undergraduate Studies in the department and to the appropriate parties at the college and university levels.

The Regent’s Policy on Student Conduct, specifically Section IV, Subd. 1. Scholastic Dishonesty, addresses these issues and can be found at [http://regents.umn.edu/sites/default/files/policies/Code_of_Conduct.pdf](http://regents.umn.edu/sites/default/files/policies/Code_of_Conduct.pdf)

Additional departmental information on academic integrity can be found here: [http://www-users.cs.umn.edu/~barry/intro/acad-conduct.html](http://www-users.cs.umn.edu/~barry/intro/acad-conduct.html)

You are expected to read and understand both of these documents.
Web Resources

- The course webpage
  
  http://www-users.cselabs.umn.edu/classes/Spring-2017/csci2041/
  
is primarily a pointer to the course Moodle page
  

where most documents and information are kept. You should have been automatically
provided access to this Moodle page; see the TA if you have not.

You are responsible for routinely checking the Moodle site for updates and announcements.

- Versions of the lecture slides shown in class will also be available on the Moodle site.

- A public repository for course information will hosted at this URL:
  
  https://github.umn.edu/umn-csci-2041S17/public-class-repo
  
This site contains descriptions for assignments and sample programs and problems.

- A course electronic discussion board will be available on the Moodle class site. The
  instructor and TAs will be reading discussions on this and providing feedback, but it will
  primarily be a forum for students to discuss course topics and questions (not solutions)
  about homework assignments.

The Disability Resource Center

The University of Minnesota is committed to providing all students equal access to learning
opportunities. The Disability Resource Center (DRC) is the campus office that works with
students who have disabilities to provide and/or arrange reasonable accommodations.

- Students who have, or think they may have, a disability (e.g. mental health, attentional,
  learning, vision, hearing, physical or systemic), are invited to contact DRC to arrange a
  confidential discussion at 612-626-1333 (V/TTY) or ds@umn.edu.

- Students registered with DRC, who have a letter requesting accommodations, are
  encouraged to discuss accommodations outlined in the letter with the instructor early in the
  semester.

Mental Health Resources

As a student you may experience a range of issues that can cause barriers to learning, such as
strained relationships, increased anxiety, alcohol/drug problems, feeling down, difficulty
concentrating and/or lack of motivation. These mental health concerns or stressful events may
lead to diminished academic performance or reduce your ability to participate in daily activities.
University of Minnesota services are available to assist you with addressing these and other
concerns you may be experiencing. You can learn more about the broad range of confidential
mental health services available on campus via http://www.mentalhealth.umn.edu.

Additional Policies

All policies (presented above) may evolve and change over the course of the semester at the
discretion of the instructor. Sometimes issues arise that cannot (or were not) planned for, and I
may need some flexibility in handling them.