CSci 2041: Advanced Programming Principles
Day and evening instances, Fall 2014

Contact information, office hours

- **Instructor:** Eric Van Wyk, email: evw@cs.umn.edu, phone: 612-625-0329.
  Office hours: Monday, Friday from 10:00am to 11:00am; or by appointment; in Keller Hall 6-203.

- **Graduate Teaching Assistants:**
  - Dan DaCosta, email: dacosta@cs.umn.edu
  - Avery Musbach, email: avery@cs.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.

- **Note:** Most course-related email should be sent to the course email alias csci2041@cs.umn.edu.

Important Dates:

- **Daytime class:**
  - The daytime class meets on Monday, Wednesday, and Friday from 11:15am until 12:05pm in Keller Hall 3-210.
  - There will be two (2) midterm exams. They are tentatively scheduled for Wednesday October 8, and Wednesday November 19.
  - The final exam is scheduled for Thursday, December 18 at 1:30pm.

- **Evening class:**
  - The evening class meets on Wednesdays from 6:30pm until 9:00pm in Akerman Hall 209.
  - There will be two (2) midterm exams. They are tentatively scheduled for Wednesday October 8, and Wednesday November 19.
  - The final exam is scheduled for Wednesday, December 17 at 6:30pm.

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.
**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; modelling 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.
**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 and collected during lectures. These may or may not be graded and are used primarily as learning 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. This work is meant to be done collaboratively; working in pairs is allowed. But each student must turn in a copy of that work.

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: 50%
  - Homework assignments – 35%
    - Different homework assignments will contribute different amounts to your cumulative homework score. This distribution will be determined as the course progresses.
    - You are expected to turn in all (outside-of-class) homework assignments if you expect 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 – 13%
    - Your lowest two lab work scores will be dropped in computing this score.
    - To pass the course you must attend at least 12 of the 15 scheduled labs, and not miss any two consecutive labs.
  - Class participation, in-class exercises, contribution to forums – 2%

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

- Cumulative exam score: 50%
  - Exam 1 – 15%
  - Exam 2 – 15%
  - Final exam – 20%

To pass the class your cumulative exam score must be a passing grade, 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 this is a new course being taught for the first time these numbers may differ. You will be informed during the semester of tentative cutoff numbers.

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

**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 and their policy on grading available at 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 should consult Moodle for this information and 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 - Cheating**

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 ranging from no credit on the work to a failing grade in the course.

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.

Software is used to detect similar assignments. When flagged, these are investigated by hand.

On the first offense, cheaters are given a 0 as their score on that assignment. This score is not counted as the lowest score to be dropped when final grades are computed; it will be used in computing your homework score.

On the second offense, cheaters are given an F 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
Additional departmental information on academic integrity can be found here: 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/Fall-2014/csci2041/) is primarily a pointer to the course Moodle page (https://ay14.moodle.umn.edu/course/view.php?id=3196) 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 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 may evolve and change over the course of the semester at the discretion of the instructor. Additional course and classroom policies are given in the first set of lecture slides.