Catalog Description
Programming emphasizing recursion, data structures, and data abstraction. Elementary analysis of and reasoning about programs. Public policy issues. Extensive programming. Three hours of lectures and one hour of computer laboratory.

Instructor Information
Prof. Susan Older (sueo@ecs.syr.edu)
Office: CST 4-181, x4679
Office hours: 3-5pm on Mondays, 1:30-3pm on Fridays, or by appointment (I’m also easy to catch after lectures)

TA Information
There will be one graduate TA and several undergraduate TAs for this course who will be helping out in the labs and possibly holding additional office hours. Information about office hours will be posted on the course web site.

More than one of us will always be available during scheduled lab sections (all in CST 1-231):
- Wed 10:35am-11:30am
- Wed 11:40am-12:35pm
- Wed 12:45pm-1:40pm
- Wed 2:15pm-3:10pm
- Wed 3:45pm-4:40pm

These lab sections will be shared across all three lecture sections of CIS 252.

Course Web Address
http://www.cis.syr.edu/courses/cis252

Textbook

Course Objectives
As its name suggests, this course introduces many of the core concepts of computer science, as well as touching on a few pieces of its history.

Broadly speaking, the practice of computer science is about data structures and supported operations on them, languages and environments, and component-based problem solving. A computer scientist must continually choose among alternatives: to do so successfully requires exposure to different options, knowledge of how to make wise choices, and experience. An important goal of this course is to expand your view of the computing landscape by helping you explore and experiment with options beyond what you’ve already seen.
We will be using the programming language Haskell for this course. Haskell is especially well suited for the two main themes we want to emphasize: patterns and problem solving. Haskell supports working with patterns in two ways. First, Haskell’s type system and extensive pattern-matching facility help a programmer write code that reflects the structure of the data it operates on. Second, using higher-order functions helps programmers capture and re-use a variety of computational patterns in a very succinct way. Haskell also supports a view of component-based design that is especially well-suited for problem solving and for logical analysis. Finally, using Haskell will require you to come to terms with recursion, a fundamental concept in computing. Consequently, you should begin to think about programming in a different, more systematic and logical way, even as you move on to imperative and object-oriented programming.

History suggests that you’ll be tempted to call this course “the Haskell course”. Our goal is that, by the end of the semester, you can see below the surface of Haskell and recognize core concepts that you will take with you to other courses and into your professional lives.

Prerequisites
There are no course prerequisites for this class. What you do need is the ability—or the willingness to develop the ability—to think precisely and analytically when it comes to problem solving.

Course Outcomes
After completion of the course, you should be proficient enough with Unix, Emacs, Haskell, and the program-design process that you can write, edit, execute, test, and debug moderate-sized programs. In addition, you should have an understanding of the Unix shell as a programming language, of Emacs as a useful program-development environment, and of Haskell as a convenient prototyping tool.

Unix and Emacs
- Use the Unix command line and other features to perform basic tasks
  - Managing files and directories
  - Checking and changing file permissions
  - Writing one-line programs using basic pipes and filters
- Use Emacs as a program-development environment

Program Design & Development
- Use a program-design recipe to develop code that is correct, coherent, and well documented.
- When given a specification of a program, construct concrete examples that satisfy that specification.
- When given a problem specification, recognize, comprehend and apply the concepts of pattern matching, recursion, and higher-order functions when designing your programs.

Haskell Fundamentals
- When given a Haskell expression, determine whether or not it is well-typed, and either give its type or explain why it is ill-typed.
- When given a Haskell type expression, give some Haskell expressions or values of that type.
- Use higher-order functions to map, filter, and fold operations across recursively defined data (e.g., lists).
- Use Haskell patterns to write data-directed programs.
- Define Haskell algebraic types to represent abstract types, and write Haskell programs that operate over those types.
• When given a moderate-sized Haskell program and relevant input, calculate the result of that program.
• When given an informal but fairly precise English description, write a Haskell program that accurately captures the desired behavior.

Other Computer Science Concepts
• When given a Turing machine description and a specific input string, be able to calculate the resulting output string.
• When given an English description of a problem, write a Turing machine that accurately captures the desired behavior.

Outcome Measurement
Your final grade will be based on a variety of activities:

• Pop Quizzes and In-class Activities (6% of final grade)
  On occasion I may give an unannounced, brief quiz or an in-class activity. These quizzes and activities will be graded on the following scale:

<table>
<thead>
<tr>
<th>Points</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Substantially correct</td>
</tr>
<tr>
<td>2</td>
<td>Partially correct</td>
</tr>
<tr>
<td>1</td>
<td>Primarily incorrect or no answer</td>
</tr>
<tr>
<td>0</td>
<td>No quiz submitted (i.e., absence)</td>
</tr>
</tbody>
</table>

These activities are intended to provide both you and me with more timely feedback on the class’s understanding of key concepts. My policy is to excuse only those absences for which I’ve received advance notification.

• Labs (12% of final grade)
  Labs provide you an opportunity to try out course concepts with us in the room to help out. There will be a lab assignment approximately every week: I will drop the lowest lab grade at the end of the semester. Each week’s lab will be made publicly available off the course web page no later than Tuesday afternoon. Your solutions are due by noon that Friday, regardless of which lab section you are in.

• Homework assignments (24% of final grade)
  Homework assignments provide you extended and more in-depth opportunities to get practice with the entire program-design process. There will be a homework assignment approximately every week: I will drop the lowest homework grade at the end of the semester.

  Homeworks assignments can be turned in at class or placed in the marked bin in the hallway near CST 4-226; you will need to complete, sign, and attach a disclosure cover sheet with each assignment.

• Exams (58% of final grade)
  Exams allow you to demonstrate your understanding and mastery of the course concepts. There will be three in-class exams during the semester. There will also be a two-hour optional final exam: the exam portion of your final grade will be the greater of (1) your cumulative average of the in-class exams, and (2) your score on the final exam.

You should also be aware of the following submission practice and policy (even more details will appear on the course web page):
On some homework and lab assignments, you may work singly or in pairs; on other assignments, you must work alone. **You may work only with students in this section of CIS 252: you should not be submitting joint work with students in Prof. Irwin’s sections.** If you work with someone else, you should submit a single solution: each person’s name must appear clearly on the first page. All members of a group will receive the same grade; groups of more than two will receive **no credit.**

Labs and homework assignments should be placed in the marked bin near CST 4-226; you will also need to submit code electronically (details to appear on the class web site). Assignments are due by the date and time specified on them: **No late assignments will be accepted.**

All grades are recorded in the Blackboard system. I encourage you to check them frequently, and let me know if you find any discrepancies. The scale for converting your numeric overall grade (calculated as stated above) to a course letter grade will be no more restrictive (see exception below) than the following:

<table>
<thead>
<tr>
<th>A numeric score of at least</th>
<th>results in a course grade of at least</th>
</tr>
</thead>
<tbody>
<tr>
<td>94.0</td>
<td>A</td>
</tr>
<tr>
<td>89.0</td>
<td>A-</td>
</tr>
<tr>
<td>84.0</td>
<td>B+</td>
</tr>
<tr>
<td>79.0</td>
<td>B</td>
</tr>
<tr>
<td>74.0</td>
<td>B-</td>
</tr>
<tr>
<td>68.0</td>
<td>C+</td>
</tr>
<tr>
<td>62.0</td>
<td>C</td>
</tr>
<tr>
<td>56.0</td>
<td>C-</td>
</tr>
<tr>
<td>50.0</td>
<td>D</td>
</tr>
<tr>
<td>0.0</td>
<td>F</td>
</tr>
</tbody>
</table>

**Important exception:** To ensure a C- or higher, your exam grade must be at least **45%**.

**Course Topics**


**Other Information**

**Academic Integrity**

All members of the Syracuse University community—faculty, staff, and students—have a duty to maintain the highest standards of academic integrity in all situations. As a member of this community, you need to be familiar with the University’s academic-integrity policy, which is available at:

http://academicintegrity.syr.edu

The university policy governs appropriate citation and use of sources, the integrity of work submitted in exams and assignments, and the veracity of signatures on attendance sheets and other verification of participation in class activities. The policy also prohibits students from submitting
the same written work in more than one class without receiving written authorization in advance from both instructors. **The presumptive penalty for a first offense by an undergraduate student is course failure**, accompanied by a transcript notation indicating that the failure resulted from a violation of the Academic Integrity Policy.

I expect all students to demonstrate high academic integrity in this course: **do not cheat, plagiarize, or commit fraud**. Fraud includes altering previously graded work or faking program transcripts to make it appear that code works correctly when it does not; plagiarism includes using someone else’s work without proper credit. If I discover any violations of academic integrity, I will give the guilty parties **failing grades (XF) for the course** and report the culprits to the Office of Academic Integrity. If you are unsure whether a certain action constitutes an academic-integrity violation, assume that it does: you may ask us for clarification at any time.

I **highly recommend** the following (inexpensive!) book to help you successfully navigate the academic-integrity waters during your collegiate career:


This book introduces the three basic tenets of academic honesty, and also provides excellent suggestions for studying for exams, working in groups, writing research papers, and other tasks you’ll encounter as a college student.

Every student must read and sign a copy of the course **Honor Policy**, which details your obligations to behave with academic integrity. Students will receive zeroes on all coursework until this sheet is turned in.

**Accommodations**

Our community values diversity and seeks to promote meaningful access to educational opportunities for all students. Syracuse University and I are committed to your success and to supporting Section 504 of the Rehabilitation Act of 1973 as amended and the Americans with Disabilities Act (1990). This means that in general no individual who is otherwise qualified shall be excluded from participation in, be denied benefits of, or be subjected to discrimination under any program or activity, solely by reason of having a disability.

If you believe that you need accommodations for a disability, please contact the Office of Disability Services (ODS), [http://disabilityservices.syr.edu](http://disabilityservices.syr.edu), located at 804 University Avenue, Room 309, or call 315-443-4498 for an appointment to discuss your needs and the process for requesting accommodations. ODS is responsible for coordinating disability-related accommodations and will issue students with documented disabilities **“Accommodation Authorization Letters,”** as appropriate. Because accommodations may require early planning and generally are not provided retroactively, please contact ODS as soon as possible.

**Computers and Other Electronic Devices**

I recognize that a small number of students like to use a computer to take notes in class. However, computers and other electronic devices also frequently become distractions, either to their owner or to others in the classroom. I ask that you put away your computer/phone/tablet/etc during lecture, unless you are using the device to take notes.

**Religious Observances** SU’s religious observances policy, found at
http://supolicies.syr.edu/emp_ben/religious_observance.htm,

recognizes the diversity of faiths represented among the campus community and protects the rights of students, faculty, and staff to observe religious holy days according to their tradition. Under the policy, students are provided an opportunity to make up any examination, study, or work requirements that may be missed due to a religious observance provided they notify their instructors before the end of the second week of classes. For fall and spring semesters, an online notification process is available through MySlice/Student Services/Enrollment/My Religious Observances from the first day of class until the end of the second week of class.