CS 381, Programming Languages, is a four-credit course for undergraduate students. This course gives an introduction to the concepts found in a variety of programming languages and to languages from a number of different paradigms. Topics to be covered are: Haskell, Prolog, scoping, parameter passing, types, polymorphism, exception handling, semantics.

Instructor:
Dr. Martin Erwig
Email: erwig@oregonstate.edu
URL: oregonstate.edu/~erwig

Office Hours:
M 11am-noon KEC 3045
W 3pm-4pm KEC 3045

Teaching Assistants:
Qiaoran Li liqiao@oregonstate.edu
M 10am-11am John 125
Parisa Ataei ataeip@oregonstate.edu
T noon-1pm KEC Atrium
Karl Smeltzer smeltzek@oregonstate.edu
W 1pm-2pm KEC Atrium
Xinyao Wang wangxiny@oregonstate.edu
W 4pm-5pm KEC Atrium

Office Hours:
M 10am-11am John 125
T noon-1pm KEC Atrium
W 1pm-2pm KEC Atrium
W 4pm-5pm KEC Atrium

Background reading material (optional):
Concepts in Programming Languages

Learning Objectives: On completion of the course, students should be able to perform the following tasks.

1. Define abstract syntax for a language that is given in concrete syntax.
2. Produce and explain the program output under static versus dynamic scoping mechanisms.
3. Produce and explain the program behavior under static versus dynamic typing mechanisms.
4. Produce and explain the program output under a selection of parameter passing mechanisms, such as by-value, by-reference, by-constant, by-result, by-value-result, and by-name.
5. Produce and explain the contents of the run-time stack as it stands at any moment in program execution.
6. Produce programs exhibiting the following kinds of polymorphism: parametric polymorphism, overloading, and subtype polymorphism, and explain their advantages and disadvantages.
7. Explain exception handling mechanisms and demonstrate the effects of exceptions on the runtime stack.
8. Explain essential differences between the imperative, functional, object-oriented, and one other programming language paradigm, and why it is important for computer science professionals to be able to understand these programming language paradigms.
9. Define the semantics of simple languages or for individual language constructs using axiomatic, operational, or denotational semantics, and given such definitions, predict specific program values or relationships between values using the definitions.
Tentative Lecture Syllabus (subject to change):

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Learning Objective</th>
<th>Quiz Tuesday</th>
<th>Quiz Thursday</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, Languages, Haskell</td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Haskell</td>
<td>8</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Abstract Syntax</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>4</td>
<td>Semantics</td>
<td>9</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Types, Polymorphism</td>
<td>3, 6</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Runtime Stack, Scoping</td>
<td>5, 2</td>
<td></td>
<td>Midterm</td>
</tr>
<tr>
<td>7</td>
<td>Parameter Passing, Exceptions</td>
<td>4, 7</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Programming Paradigms, Prolog</td>
<td>8</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Prolog</td>
<td>8</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Prolog</td>
<td>8</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Important dates

<table>
<thead>
<tr>
<th>Quizzes</th>
<th>Apr 10</th>
<th>Tue</th>
<th>2pm - 2:20pm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Apr 19, 26</td>
<td>Thu</td>
<td>2pm - 2:20pm</td>
</tr>
<tr>
<td></td>
<td>May 3</td>
<td>Thu</td>
<td>2pm - 2:20pm</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>May 10</td>
<td>Thu</td>
<td>2pm - 3:00pm</td>
</tr>
<tr>
<td></td>
<td>May 17, 24, 31</td>
<td>Thu</td>
<td>2pm - 2:20pm</td>
</tr>
<tr>
<td>Final Exam</td>
<td>June 11</td>
<td>Mon</td>
<td>2pm - 3:50pm</td>
</tr>
</tbody>
</table>

Grading

20% Quizzes
20% Homework
25% Midterm Exam
35% Final Exam

Note on Quizzes & Exams: You are allowed to use a one-page cheat sheet during each quiz or exam (single-sided, at least 10pt font size). No other supporting devices (books, cell phones, computers, etc.) are allowed.

Classroom Policies

We have about 170 students in this class. In support of a focused learning experience for everyone, I ask you to adhere to the following rules during lectures.

- **Don't use any electronic devices** except for Piazza polls or programming exercises.
- **Don't eat** during class.
- **It is prohibited** to create any photo, audio, or video recordings of lectures.

Note on Homework: Teamwork on homework is allowed. Teams of up to five students can submit a common homework. All team members must be clearly identified on the submitted homework. All team members will receive the same points for that homework. All students in a team must contribute to a team solution. Just adding the name of a student who has not contributed to a solution is not acceptable. Such cases will be regarded as cheating and will be penalized. All team members must be able to explain their homework contribution to the instructor.

Students with documented disabilities who may need accommodations, who have any emergency medical information the instructor should know, or who need special arrangements in the event of evacuation, should make an appointment with the instructor as early as possible, however, no later than the first week of the term. In order to arrange alternative testing the student should make the request at least one week in advance of the test. Students seeking accommodations should be registered with the Office of Services for Students with Disabilities.