# CS 381—Programming Language Fundamentals

Syllabus for Winter 2015

---

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:**
Karl Smeltzer
Email: smeltzek@eecs.oregonstate.edu

**Office Hours (KEC 3119)**
Monday 14:00–15:00
Wednesday 14:00–15:00
***or by appointment***

**Teaching Assistants:**

| | | | | |
|---|---|---|---|---|
| Keeley Abbott | abbottk@onid.oregonstate.edu | Tue | 8:30–9:30 | DEAR 203 |
| Chao Peng | pengc@onid.oregonstate.edu | Fri | 10:00–11:00 | KEC Atrium |

**Special Note:** This is my first time teaching a full course, which means that there are going to be a few hurdles along the way. I ask that you please help make the class better for everyone by contacting me directly and immediately (as opposed to waiting for evaluations) if you have any thoughts or concerns about improving class policies, course material, teaching style, etc. so that I can make appropriate changes as quickly as possible.

**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):**

| Week | Topic | Learning Objective | Quiz | | |
|---|---|---|---|---|---|
| | | | Monday | Wednesday | Friday |
| 1 | Introduction, Languages, Haskell | 8 | | | |
| 2 | Haskell | 8 | | ✓ | |
| 3 | Abstract Syntax | 1 | No class | | |
| 4 | Semantics | 9 | | ✓ | |
| 5 | Types, Polymorphism | 3, 6 | | ✓ | |
| 6 | Runtime Stack, Scoping | 5, 2 | | Midterm | |
| 7 | Parameter Passing, Exceptions | 4, 7 | | ✓ | |
| 8 | Programming Paradigms, Prolog | 8 | | ✓ | |
| 9 | Prolog | 8 | | | ✓ |
| 10 | Prolog, Review | 8 | | | |

**Important dates (subject to change)**

| | | | |
|---|---|---|---|
| ***Quizzes*** | **Jan 14** | Wed | 13:00–13:50 |
| | **Jan 23, 30** | Fri | 13:00–13:50 |
| | **Feb 6** | Fri | 13:00–13:50 |
| ***Midterm Exam*** | **Feb 11** | Wed | 13:00–13:50 |
| ***Quizzes*** | **Feb 18, 25** | Wed | 13:00–13:50 |
| | **Mar 6** | Fri | 13:00–13:50 |
| ***Final Exam*** | **Mar 16** | Mon | 14:00–15:50 |

**Grading**

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

**Note on Exams:**
All exams are ***closed book***.

**Classroom Policies**

To support a more focused learning experience for everyone, please adhere to the following rules during lectures.

- Please don't use cell phones (including texting) during class.
- Please don't eat during class.
- Please use laptops, tablets, etc. in class ***only*** for taking notes or while working on exercises.

**Note on Homework:** Teamwork on homework is *allowed*. Teams of two or three students can submit a common homework so long as all members are clearly identified on the submission. All students in a team must contribute to a team solution and will receive the same grade. Just adding the name of a student who has not contributed to a solution 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.**