CS 381 – Programming Language Fundamentals

Winter 2019 – Syllabus

CS 381, Programming Languages Fundamentals, is a four-credit course for undergraduate students. The course introduces concepts found in a variety of programming languages and exposes students to non-imperative programming paradigms. Topics to be covered include: functional programming in Haskell, logic programming in Prolog, syntax, scoping, parameter passing, types, polymorphism, exception handling, semantics.

Course Staff and Logistics

Instructor   Eric Walkingshaw (walkiner@oregonstate.edu)

Grad TAs   Ghadeer Al Kubaish (alkubaig@oregonstate.edu)
Divya Bajaj (bajajd@oregonstate.edu)
Qiaoran Li (liqiao@oregonstate.edu)

Lectures   Mon/Wed/Fri 11:00–11:50am, Withycombe Hall 109

Web page   http://web.engr.oregonstate.edu/~walkiner/teaching/cs381-wi19

Office hours   (see course web page)

Learning Objectives

At the end of the course, students should be able to . . .

1. **Design and implement** a functional program, using *algebraic data types* and *recursive functions*, from an informal specification and example input-output pairs.
2. **Produce and explain** the *type* and *result* of an expression within a larger functional program.
3. **Produce and explain** an *abstract syntax*, using a context-free grammar, for a language given a definition of its concrete syntax.
4. **Analyze and reduce** a language, by applying the concept of *syntactic sugar*, to its minimal core of essential features.
5. **Produce and explain** how a program's output and behavior differs based on a variety of language design decisions related to naming, such as *static vs. dynamic scope* and *call-by-value vs. call-by-name vs. call-by-need* parameter passing.
6. **Design and implement** a *denotational semantics* or *interpreter* for a language given its abstract syntax and an informal specification of its behavior, which may include features such as naming, state, functions, and/or exceptions.
7. **Design and implement** a logic program, using *queries* over recursively defined *predicates*, from an informal specification and example results.

Tentative Schedule (subject to change)

Week	Topic	!!!
1	Introduction, Haskell	
2	Haskell	
3	Haskell, Syntax	Quiz #1
4	Syntax	Midterm #1
5	Semantics	
6	Types	
7	Naming, Parameter Passing	Quiz #2
8	State, Exceptions	Midterm #2
9	Prolog	
10	Prolog, Review	Quiz #3

Final exam: Wed Mar 20, 12:00–1:50pm

For the latest scheduling information, check the course web page regularly!

Grading

Grades will be computed using the following weights:

Homework	25%
Quizzes	15% = 3 × 5% each
Midterms	30% = 2 × 15% each
Final Exam	30%

Grades are assigned using the following ranges: $\geq$93% = A, 90–92% = A-, 87–89% = B+, 83–86% = B, 80–82% = B-, 77–79% = C+, 73–76% = C, 70–72% = C-, 60–69% = D, $\leq$59% = F. Overall grades will not be curved. Scores on individual quizzes and exams may occasionally be adjusted upward by a constant factor for the entire class.

Course Policies

All **quizzes and exams** are closed book and closed notes.

Teamwork on **homework** is allowed and encouraged. Teams of two or three students may 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. Adding the name of a student who has not contributed to a solution will be regarded as cheating. Submitting work that is not yours will also (obviously) be regarded as cheating. All team members must be able to explain their homework contribution to the instructor, if asked.

Students with Disabilities

Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS). Students with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098.