

COMP 210

Principles of Computing and
Programming (Fall 2007)



Prof. Walid Taha

Department of Computer Science

Rice University



Instructor Information

- Walid Taha
- Duncan Hall (DH) 3103
- taha@rice.edu
- www.cs.rice.edu/~taha
- (713) 348 5718 - *but email is better!*
- Office hours:
 - MWF 11:00-11:55, and
 - by appointment



Course Materials

- Official course page:
 - www.owlnet.rice.edu/~comp210/
- Actual course page:
 - www.hope.cs.rice.edu/twiki/bin/view/Teaching/210
- ... but all you really need to remember is
 - Google for “**comp 210**” and you’ll find it.



Why Programming?

- Programming is essential to understand *computing*
- Computing is essential for individual and national competitiveness and well-being
- Computing has a dramatic impact on our life
 - “Code is Law” [Lessig]
 - Talked to any customer service reps lately?
 - Medical information, financial information, libraries, historic records, voting
 - Embedded and real-time systems
 - Cars, phones, airplanes, iRobot, iPhone, ...
- Computing is *intellectually challenging*



Why COMP 210?

- Innovative curriculum
 - Developed at Rice, with NSF funding
- How it is different
 - Extreme focus on core skills
 - Very few programming constructs (in Scheme)
 - **Data types** drive the problem solving process
 - Data types are **not** “data structures”
 - Not your typical “programming course”, and
 - Not a “Scheme language course”
 - This is an intro to computing, **not** coding



Course Overview

- Data-driven program design 2-17
- Abstraction and good design 19-23
- Algorithms 25-28
- Accumulators 30-32
- Side effects 34-37
- Mutable structures, and objects 39-43

- Design Recipes
 - Introduced and used throughout the course



Problems You Learn to Solve

- Area of different shapes -> filtering lists 2-17
- Identifying and solving “problem families” 19-23
- Graph algorithms (used in games and planning) 25-28
- Non-data-type-driven problems (?) 30-32
- Problems that modify state 34-37
- Problems involving complex objects 39-43

- Design Recipes
 - We show more of the recipe as we see more interesting problems



Grading

- Homeworks (worth 40-45%)
 - Usually once a week, Monday-to-Monday,
 - Work in teams of two. Do **not** divide work up.
 - No late homework will be accepted
- Exams (worth 50%) Prior exams available online.
 - In class, closed book, closed notes.
 - 10% for first, 20% for second, 20% final
- Quizzes (worth 5-10%)
 - You get 7/10 just for taking the quiz. It helps **you**.
- Your performance in each part **must** be consistent.



Why Scheme

- Simple syntax, and we will use only part of it:
 - Function definition
 - Function application
 - Conditionals
 - Structure definitions
 - Local definitions
 - Assignment (much later in the course)
- Simple semantics
 - Can be understood abstractly: It doesn't depend on a “vendor”
 - Can be understood rigorously: to check our program is **correct**
 - Nice integrated design environment (IDE)
- Functional languages help **reasoning** and **concurrency**



How to Succeed

- Do the reading before class
 - This will help you on the quiz
- Attend class
 - Both reading AND lectures are required.
 - Exam questions will emphasize any low-attendance classes
- Take homework assignments seriously
 - A working program is worth **Zero Points**
- Use office hours
 - Having questions is a sign of intelligent life



COMP 210 Prepares You to...

- Learn other languages, like Java, without getting bogged down in the details.
- Learn deeper concept of computing:
 - Algorithms (including artificial intelligence techniques, data-mining, bioinformatics)
 - Formal methods (programming languages, verification, formal logic)
 - Systems (networks, operating systems, compilers)
 - Architecture (digital logic, microprocessor design)



About Your Instructor

- My research program is concerned with
 - Improving programmer productivity, using
 - Automatic program generation,
 - Lightweight formal verification (type systems),
 - Functional languages (Scheme, ML, Haskell).
 - Improving productivity of people building:
 - Real-time and embedded systems,
 - Hardware (microprocessors or “chips”),
 - Demonstrating the benefits of machine-checked formal proof



Copyright

- COMP 210 lecture notes are copyrighted by Prof. Walid Taha, Rice University.
- COMP 210 students can copy and modify these materials freely as long as this slide is preserved
- Commercial use requires written permission.
- Publicly available materials for the course are licensed under the Creative Commons (CC) license
 - See creativecommons.org basic idea, and then course Twiki for details



Next Lecture

- Make sure you've done Homework 0
 - Already posted online on web-page
 - Due next class (Wednesday)
- Next class
 - The basics of programming