S11

Online Book	**Owlspace Course Site**	**Office Hours**	**Turnin Guide**	**Scheme HW Guide**	**Java HW Guide**	**HW Checklist**	**HW Grading**

COMP 211: Principles of Program Design

Instructors:	Prof. Robert "Corky" Cartwright	**Staff:**	Alina Sbirlea
	Dr. Stephen Wong		Kamal Sharma
			Nicholas Coltharp
Lectures:	Duncan Hall (DH) 1075	**Time:**	MWF 10:00-10:50am
Lab:	Ryon 102	**Time:**	Tuesday 10:50am-12:05pm

Office Hours:

- Dr. Cartwright: MWF 9:00 AM - 10:00 AM
- Dr. Wong: See home page
- Alina: MWF 11:00 AM -12:00 PM
- Kamal: MF 3:00 PM - 5:00 PM
- Nicholas: TR 7:00 PM - 8:30 PM

e-mail the entire class: comp211s11 at owlspace-ccm.rice.edu

e-mail just the staff: comp211 at rice.edu

Introduction

This course is an introduction to the fundamental principles of programming. The focus is on systematic methods for developing robust solutions to computational problems. Students are expected to have experience writing interesting programs in some credible programming language (e.g., Python, Java, Scheme, C#, C++, Visual Basic .NET, PRL, Scheme, Lisp, etc.) but no specific programming expertise is assumed. The course is targeted at potential Computer Science majors but mathematically sophisticated non-majors are welcome. We expect students to be comfortable with high-school mathematics (primarily algebra, mathematical proofs, and induction) and the mathematical rigor and vocabulary of freshman calculus. Success in the course requires a deep interest in the foundations of computer science and software engineering, self-discipline, and a willingness to work with other people on programming projects. Topics covered include functional programming, algebraic data definitions, design recipes for writing functions, procedural abstraction, reduction rules, program refactoring and optimization, object-oriented programming emphasizing dynamic dispatch, OO design patterns, fundamental data structures and algorithms from an OO perspective, simple Grapical User Interfaces (GUIs), and an exposure to the challenges of concurrent computation.

Students will learn the practical skills required to write, test, maintain, and modify programs. Labs and assignments use the Scheme and Java programming languages.

Text For Scheme: How to Design Programs by Felleisen et al. QA76.6 .H697 2001 (Available online; no purchase is necessary.)

DrScheme/DrRacket: Please download and use the DrScheme/DrRacket system available from the Racket download site. To avoid compatibility problems, please make sure you use a Version numbered 5.0.2

- DrRacket Tips and Traps

Notes for Java: Object-oriented Design

References for Java

- Principles of Object-Oriented Programming by Zung Nguyen and Stephen Wong. An online self-contained introduction to OOP in Java roughly corresponding to the former Comp 212 course. It is reasonably complete, but still under construction.
- Design Patterns Lens - A collection of short descriptions of the design patterns covered in the course.
- Index to online Java Tutorials by Sun Microsystems The Sun tutorials refer to the full Java language not the Elementary and Intermediate language levels supported by DrJava.Nevertheless, they cover many important language details in depth, such as the complete collection of primitive operators on primitive data types.
- Java Basics by Fred Swartz is a clearly written traditional introduction to Java that focuses on Java mechanics rather than OO Design. It can be helpful in learning the *mechanics* of writing full Java code. Please ignore what he says about program design.
- Java Notes by Fred Swartz is a reasonably comprehensive Java reference that is a good supplement to the official Sun documents.

DrJava: Please download and use the DrJava pedagogic programming environment available from drjava.org. You must install either the Java 5 or Java 6 JDK on your machine for DrJava to work. (Addendum: DrJava releases starting in 2011 can also run using a Java 7 JDK.) If you machine is running some flavor of Windows or Linux, go to the \[Sun Download Site for the Java SE 6.0 (http://java.sun.com/javase/downloads/index.jsp\]). Make sure that you download the **JDK** not the **JRE**. If you have a Mac, a Java JDK is available from Apple. In fact, it is part of the standard Mac OS X software package. DrJava will run on either the Apple Java 5.0 or Java 6.0 JDK.

Entrance Survey

Please fill out this survey.

Confirm that You Can Attend Lab

The lab is an essential component of the course. Every student should attend lab on Tuesday 10:50-12:05 in Ryon 102.
If you have a conflict, you probably should drop the course. If you have a conflict, but still want to take the course, please come see Prof. Cartwright or Prof. Wong to evaluate your options.

Computing Environment

All Comp 211 programming assignments will be run and graded on the CLEAR (Linux) educational computing facility. The Ryon 102 lab is equipped with machines connected to the CLEAR network. For instructions on how to use CLEAR, see the CLEAR web page.

Errata

This wiki for Comp 211 is a revision of last year's and may contain a significant number of broken links and typos. If you notice an error in the wiki, please send email to the Comp 211 staff.

Course Schedule

Note that future date schedules are only guidelines. Future homeworks and slides may contain materials from previous Comp 210 and Comp 212 classes. New material will be provided before the corresponding class. There will only be two exams in the course: one given on functional programming during week 7 and one on object-oriented programming given during in the last week of the course. Both are take-home exams. There is no final examination.

	Day	Date (2011)	Topic	Reading	Lectures	Problems	Due(2011)	Lab	Supplements
1	Mon	Jan 10	Introduction & Scheme Primitives		L 1	HW 0	W Jan 12	Lab 0	Pair Programming
2	Wed	Jan 12	Function definitions and conditionals	Read Chs. 1-10	L 2	HW 1	F Jan 21		
3	Fri	Jan 14	Data Definitions & The Design Recipe	Read Chs. 11-13	L 3				
-	Mon	Jan 17	School Holiday					Lab 1	
4	Wed	Jan 19	Data-directed design	Read Chs. 14-15	L 4				
5	Fri	Jan 21	Data-directed design: trees		L 5	HW 2	Mon Jan 31		
6	Mon	Jan 24	Mutually Referential Data Definitions	Read Chs. 16-17	L 6			Lab 2	Class Demo
7	Wed	Jan 26		Read Ch. 18	L 6				
8	Fri	Jan 28	Local Definitions and Lexical Scope	Read Chs. 19-20	L 7				
9	Mon	Jan 31	Functional Abstraction and Polymorphism	Read Chs. 21-22	L 8	HW 3	Mon Feb 7	Lab 3	
10	Wed	Feb 02	Functions as Values	Read Ch. 24	L 9				
11	Fri	Feb 04	Lambda the Ultimate	Read Chs. 25-28	L 10				
12	Mon	Feb 07	Generative Recursion	Study Chs. 25-28	L 11	HW 4	Mon Feb 14	Lab 4	
13	Wed	Feb 09	Complexity and Accumulators	Read Chs. 29.1-2 Skim Chs. 30-32	L 13				
14	Fri	Feb 11	Accumulators and Tail Calls	Read Chs. 30-32	L 14				
15	Mon	Feb 14	Clever Programming With Functions	Review prior readings	L 15	HW 5	Mon Feb 21	Lab 5	210 Exam 1 210 Exam 2 Solution to 12.4.2
16	Wed	Feb 16	Exam Review	Review prior readings	L 16, L 17				
17	Fri	Feb 18	On to Java	OO Design Notes Ch 1.1-1.4	L 18				
18	Mon	Feb 21		OO Design Notes Ch 1.1-1.4	L 18	HW 6 (Optional) Exam 1	Mon Mar 07	Lab 6	
19	Wed	Feb 23	Java Design Recipe	OO Design Notes Ch 1.1-1.4	L 19				IntList IntListTest

20	Fri	Feb 25	Defining Inductive Data in Java	OO Design Notes Ch 1.5 Tony Hoare: "Null References: The Billion Dollar Mistake"	L 20				
-	M-F	Feb 28-Mar 4	Spring Break						
21	Mon	Mar 07	Static Class Members and the Singleton Pattern	OO Design Notes Ch 1.6	L 21	HW 7	Mon Mar 14	Lab 7	ObjectList ObjectListTest
22	Wed	Mar 09	Polymorphism and Interfaces	OO Design Notes Ch 1.8	L 22				ComparableList ComparableListTest
23	Fri	Mar 11	Handling Exceptions and Errors	OO Design Notes Ch 1.9-1.10, 1.12	L 23				
24	Mon	Mar 14	The Strategy and Visitor Patterns	OO Design Notes Ch 1.9, 1.11	L 24	HW 8	Mon Mar 21	Lab 8	IntList IntListVisitor IntListTest
25	Wed	Mar 16	Visitors, Visitors, Vistors ...	OO Design Notes Ch 1.11	L 25				
26	Fri	Mar 18	Accepting Reality: Full Java	OO Design Notes Ch 1.13	L 26				IntList IntListTest
27	Mon	Mar 21		OO Design Notes Ch. 1.10, 1.13		HW 9	Fri Apr 4	Lab 9	List ListTest
28	Wed	Mar 23							
-	Thurs-Fri	Mar 24-25	School Holiday						
29	Mon	Mar 28	Simple Generics in Java	OO Design Notes Ch 1.13.4 (contains advanced material as well)	L 29			Lab 10	lec29_code.zip (incl. genList)
30	Wed	Mar 30	Mutation and Bi-Directional Linked Lists	OO Design Notes Ch 1.13	L 30				
31	Fri	Apr 1	BiLists Continued	OO Design Notes Ch 2.1	L 30				BigBiList BigBiListTest
32	Mon	Apr 4	Mutable Trees	OO Design Notes Ch 2.1	L 32	HW 10	Wed. Apr 13	Lab 11	TreeMap TreeMapTest OOTreeMap OOTreeMapTest
33	Wed	Apr 6	Mutable Trees		L 32				
34	Fri	Apr 8	Mutable Trees		L 32				FunctionalQuicksort
35	Mon	Apr 11	Mutable Trees and OO Data Structure Review	OO Design Notes	L 32			Lab 12	
36	Wed	Apr 13	OO Sorting Algorithms	CNX Module on Sorting (incl. insertion and selection sort animations)	L 36	HW 11	Milestone 1: Mon. Apr 18 Milestone 2: Fri. Apr 22		Design Patterns Slides Design Patterns Paper Sorter Demo
37	Fri	Apr 15	Graphical User Interfaces		L 37, 2010				
38	Mon	Apr 18	Fast Searching and Memoization		L 38			Lab 13	MyHashMap.java MyHashMapTest.java BetterMath.java
39	Wed	Apr 20	Fast Sorting Methods		L 39				
40	Fri	Apr 22	Review		L 40				

Grading, Honor Code Policy, Processes and Procedures

Grading will be based on your performance on homeworks (worth 50%) and exams (20% for first exam, and 30% for the second exam).

Take-home exams, which are pledged under the honor code, test your individual understanding and knowledge of the material. Collaboration on exams is strictly forbidden.

We will normally use the OwlSpace Comp 211 Announcements and General Discussion forum to post important announcements and discussions related to the
class, particularly Q&A on class assignments. The Owlspace e-mail alias to the entire class (see above) will also be used.

Other Mailing Lists:

- cs-events-l@mailman.rice.edu:
  - Announcements relating to talks and other interesting events hosted by the CS departments.
  - Subscription to this list is optional but highly recommended

Questions

If you have a question about homework – you're not sure what is expected for a given problem, you haven't received feedback from a previous assignment, or you don't understand or agree with the assessment of your work, for example – you can raise the question with a TA in lab or on the class mailing list. It is preferable to send a question out for the whole class to discuss so that everyone can benefit from eachother's contributions as well the responses from the staff.   Please restrict messages sent to only the staff to more sensitive matters that do not warrant viewing by the entire class. If you still don't feel that your concerns have been addressed, you may wish to contact Prof. Cartwright or Prof. Wong directly.

Homeworks:

Homeworks help you verify your understanding of the material and prepare you for the exams. You are encouraged to discuss the homework problems with the instructors and staff. Help from other students, including Comp 211 graduates, is also encouraged (but should be cited), although that does *not* include giving or receiving complete answers. All homework partners are responsible for knowing all the submitted material. If you fail to understand the homework solutions, you won't succeed on the exams.

Homeworks will generally be handed out on Fridays, and will be due before class the following Friday.

You are expected to work in groups of two.

You may change partners during the semester.

Partners should work together on all aspects of the homework -- all students are expected to contribute equally. You and your partner should hand in exactly one solution.

Late homework will not be accepted with one exception. Every student is allotted 7 slip days. Each whole day or fraction of a day that an assignment is late counts as a slip day. Each student in the pair submitting a late assignment must spend the requisite number of slip days. Since assignments get progressively harder during the semester, we strongly encournage you to hoard your slip days for use near the end of the term.

We recommend that you review the homework guide as you develop your solutions. Review the submission checklist when you turn in your homework. Your work will be graded as documented on the grading page.

Reading: For each lecture, there is associated reading. Students are required to complete the reading before the class associated with this reading.

Other Resources

- Practical matters:
  - DrScheme Programming Environment
- Special interest groups:
  - CSters
  - Computer Science Club
  - ... (please send in suggestions!)

Additional References

Here is a nice article about the basic approach taken in this course.

More on CS

The New Turing Omnibus, A. K. Dewdney	QA76 .D448 1993
Algorithmics: The Spirit of Computing, David Harel	QA76.9 .A43 H37 2004
Computers Ltd.: What They Really Can't Do, David Harel	QA76.5 .H3575 2000
Great Ideas in Computer Science, Alan W. Biermann	QA76 .B495 1997
Computer Science: An Overview, J. Glenn Brookshear	QA76 .B743 1997
Godel, Escher, Bach: An Eternal Golden Braid, Douglas Hofstadter	QA9.8 .H63 1980
Metamagical Themas, Douglas Hofstadter	Q335 .H63 1985

More on Functional Programming

Google's MapReduce	(Online)
The Little Schemer, Friedman & Felleisen	QA76.73 .S34 F75 1996
The Seasoned Schemer, Friedman & Felleisen	QA76.73 .S34 F77 1996
Developing Applications with Objective Caml, Emmanuel Chailloux, Pascal Manoury, and Bruno Pagano	
The Haskell School of Expression: Learning Functional Programming Through Multimedia, Paul Hudak	QA76.62 H83 2000

More on Object-Oriented Design in Java

Principles of Object-Oriented Programming	The Connexions Curriculum for the former Comp 212 course
The Java Programming Language, Arnold & Gosling	Gosling was the principal designer of Java 1.0
Thinking In Java, Bruce Eckel	The 3rd edition is available free.
Design Patterns: Elements of Reusable Object-Oriented Software, Gamma, Helm, Johnson & Vlissides	The "Bible" of OO software design.
Head First Java, Bert Bates & Kathy Sierra	A fun read while you get introduced to Java and learn how to think like a Java Programmer.
Head First Design Patterns, Freeman & Freeman	An accessible introduction to Design Patterns.

More on Data Structures and Algorithms

Introduction to Algorithms: Book written by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest. The authoritative reference on algorithms.

Accommodations for Students with Special Needs

Students with disabilities are encouraged to contact me during the first two weeks of class regarding any special needs. Students with disabilities should also contact Disabled Student Services in the Ley Student Center and the Rice Disability Support Services.