

CS151.02 2007F At A Glance

This is an abbreviated course syllabus. Like everything else in this course, it is likely to change.

Weeks: 1, 2, 3, 4, 5, 6, 7, break, 8, 9, 10, 11, 12, 13, 14.

	Date	Topic	Reading	Lab	Assignments
<i>Week 1: Getting Started</i>					
01	Thursday, 30 August 2007	An Introduction to Algorithms Introduction: What is CS? Exercise: Drawing smileys.		Drawing Smiley Faces	
02	Friday, 31 August 2007	An Introduction to CSC151 Lessons from day one. Common parts of an algorithm. About the course. Some administrative details. Getting started with Linux.	Grinnell's Linux Environment	Getting Started with Linux	Assignment 1: Class Basics
03	Monday, 3 September 2007	An Introduction to Scheme Why use programming languages? Scheme basics. Scheme history.	The DrFu Program-Development Environment & Beginning Scheme	The DrFu Program-Development Environment	
04	Tuesday, 4 September 2007	Raster Graphics Representing images, revisited. Pixels and colors: The basics. The graphics coordinate system.	Raster Graphics: Images from Pixels and Colors	Raster Graphics	Assignment 2: Drawing Smileys, Revisited

05	Thursday, 6 September 2007	RGB Colors Problem: representing colors. The RGB representation. Complementary colors.	RGB Colors	RGB Colors	
06	Friday, 7 September 2007	Transforming Colors Computing new colors from old. Some basic transformations. Transforming pixels and images. Composing transformations.	Transforming RGB Colors	Transforming RGB Colors	Assignment 3: Algorithmic Image Summaries
Week 2: Writing Functions					
07	Monday, 10 September 2007	A Design Perspective Approaching colors. Managing the huge palette. Other design issues.	<i>None</i>	<i>None</i>	
08	Tuesday, 11 September 2007	Transforming Images Review: Transforming colors. Expanding transformations with map. Sequencing transformations with compose.	Transforming Images	Transforming Images	Assignment 4: Blending
09	Thursday, 13 September 2007	Writing Your Own Color Transformations What is a procedure? Describing procedures. Syntax: Writing procedures.	Anonymous Procedures	Writing and Using Anonymous Color Transformations	
10	Friday, 14 September 2007	Computing with Numbers Types. Kinds of Numbers. Key Numeric Operations.	Numeric Values	Writing More Complex Color Transformations	Assignment 5: Colors in Context

Week 3: Control					
11	Monday, 17 September 2007	Iterating Over Positions Iteration, a key aspect of algorithm design. Iterating over positions. Blends and other positionally-computed images. Simulating <code>image.map!</code> .	Building Images by Iterating Over Positions	Building Images by Iterating Over Positions	
12	Tuesday, 18 September 2007	Boolean Values and Predicate Procedures Boolean values. Predicates - Procedures that return Boolean values. Combining booleans with <code>and</code> and <code>or</code> .	Boolean Values and Predicate Procedures	Boolean Values and Predicate Procedures	Assignment 6: Creating Filters
13	Thursday, 20 September 2007	Conditionals Choosing between two options with <code>if</code> . Making multiple choices with <code>cond</code> .	Conditionals	Conditionals	
14	Friday, 21 September 2007	Conditionals, Revisited Drawing with conditionals. Debriefing: <code>if</code> vs. <code>cond</code> .		Conditionals	Assignment 7: Overlapping Bordered Gradients
Week 4: Lists and Recursion					
15	Monday, 24 September 2007	Representing Images as Lists of Spots Spot lists: Another perspective on images. Lists in Scheme. Basic list operations. Processing spot lists.	Representing Images as Lists of Spots	Representing Images as Lists of Spots	

16	Tuesday, 25 September 2007	Iterating Over Lists Building new lists from old with map. Doing something with each value in a list with foreach!. Drawing lists of pixels.	Iterating Over Lists		
17	Thursday, 27 September 2007	List Iteration Lab Iterating lists of spots.	Iterating Over Lists	Iterating Over Lists	
18	Friday, 28 September 2007	Recursion Basics The idea of recursion. A sample recursive procedure: sum. Another example: Filtering.	Recursion Basics	Recursion Basics	Exam 1: Scheme Basics
Week 5: Recursion, Revisited					
19	Monday, 1 October 2007	Tail Recursion The key idea of recursion. A new technique: Passing along intermediate results. Special case: Tail recursion.	Tail Recursion	Tail Recursion	
20	Tuesday, 2 October 2007	Recursion, Revisited Q&A. Optional topics: Scheme's Evaluation Strategy; Another Example.	Recursion, Revisited	Recursion, Revisited	Assignment 8: Building Lists of Spots
21	Thursday, 4 October 2007	Recursion, Revisited, Revisited More practice with recursion.	Recursion, Revisited	Recursion, Revisited	
22	Friday, 5 October 2007	Pause for Breath (1): An Overview of Scheme			Assignment 9: Recursion
Week 6: Miscellaneous					

23	Monday, 8 October 2007	Pause for Breath (2): A Review of Procedures What is a procedure? Describing procedures in Scheme. Naming procedures in Scheme. How Scheme evaluates procedures.			
24	Tuesday, 9 October 2007	Pause for Breath (3): A Review of Recursion Basic concepts of recursion. Patterns of recursion. Selected examples.			
25	Thursday, 11 October 2007	Documenting Programs and Procedures The need for documentation. The Six P's - a strategy for documenting procedures. Practice.	Documenting Your Procedures		
26	Friday, 12 October 2007	Preconditions and Postconditions Verifying preconditions. Husk and Kernel programming.	Verifying Preconditions	Verifying Preconditions	Assignment 10: Documentation
Week 7: Miscellaneous					
27	Monday, 15 October 2007	Testing Your Procedures Why test? Strategies for testing. The primary testing operations.	Testing Your Procedures	Testing Your Procedures	

28	Tuesday, 16 October 2007	Analyzing Procedures Comparing algorithms. Two related metrics: Time and Number of steps. Counting procedure calls by printing. Tools for analysis.	Analyzing Procedures	Analyzing Procedures	
29	Thursday, 18 October 2007	Numeric Recursion Recursion, Generalized. Thinking About Natural Numbers. Numeric Recursion.	Numeric Recursion	Numeric Recursion	
30	Friday, 19 October 2007	Tools for Programming the Gimp Another model of images. Drawing through selection. Other useful techniques.	GIMP Tools	Lab: GIMP Tools	Assignment 11: Testing
Fall Break!					
Week 8: Miscellaneous					
31	Monday, 29 October 2007	Randomized (Unpredictable) Drawings Why use randomness. The random procedure. Random art.	Randomized Drawing	Randomized Drawing	
32	Tuesday, 30 October 2007	Geometric Art Through Numeric Recursion Parallel lines, Concentric Circles, and Beyond.	Geometric Art	Geometric Art	

33	Thursday, 1 November 2007	Naming Local Values Why name things. Naming things with <code>let</code> . Naming things with <code>let*</code> . Naming procedures. Lab.	Local Bindings	Local Bindings	
34	Friday, 2 November 2007	Husk and Kernel, Revisited Husk and Kernel Programming, Revisited. An example. Preventing clients from calling the kernel.			Assignment 12: Spirograph-Like Drawings
Week 9: Files					
35	Monday, 5 November 2007	Naming Local Procedures Why have local procedures. Creating local procedures.	Local Procedure Bindings	Local Procedure Bindings	
36	Tuesday, 6 November 2007	Characters and Strings Representing text. Characters: The basic building blocks. Combining characters into strings.	Characters and Strings	Characters and Strings	
37	Thursday, 8 November 2007	File Basics Data persistence. Basic file operations.	Files	Files	
38	Friday, 9 November 2007	Storing Images as Simple Pixel Maps Storing images. Storing colors. Storing images, revisited.	Pixel Maps: A Technique for Storing Images	Pixel Maps	Exam 2: Recursion and Beyond
Week 10: Other Data Structures					

39	Monday, 12 November 2007	Pixel Maps, Revisited Storing data. Tradeoffs.	Pixmap, Revisited: Encoding Data More Efficiently	Pixmap, Revisited	
40	Tuesday, 13 November 2007	Color Palettes Saving space with palettes. Choosing palette colors. Using different palettes.	Color Palettes	Color Palettes	Assignment 13: Saving Shape-Based Drawings in Files
41	Thursday, 15 November 2007	Representing Color Palettes with Vectors Problems with lists. A solution: Vectors. Important vector procedures.	Vectors	Vectors	
42	Friday, 16 November 2007	Pairs and Pair Structures Memory and naming. Pairs and cons cells. Why care?	Pairs and Pair Structures	Pairs and Pair Structures	Assignment 14: A Procedure Is Worth A Thousand Pictures
<i>Week 11: Algorithms</i>					
43	Monday, 19 November 2007	Deep Recursion, Revisited Lists, revisited. Fractals, reviewed. Trees, introduced. Deep recursion, considered.	Deep Recursion	Deep Recursion	
44	Tuesday, 20 November 2007	Association Lists and Searching Storing information in tables. Representing table entries as lists. Representing tables as lists. Association list: Scheme's standard table representation. Implementing key alist procedures.	Association Lists	Association Lists	Assignment 15: Fractals, More or Less
<i>Thanksgiving Break!</i>					
<i>Week 12: Algorithms and Data Structures</i>					

45	Monday, 26 November 2007	Higher Order Procedures Procedures as parameters. Procedures as return values. Writing map. Writing all?.	Higher-Order Procedures	Higher-Order Procedures	
46	Tuesday, 27 November 2007	Binary Search Analyzing algorithmic efficiency. Making searching more efficient. Balanced trees.	Searching	Binary Search	
47	Thursday, 29 November 2007	Introduction to Sorting The problem of sorting. Writing sorting algorithms. Examples: Insertion, selection, etc. Formalizing the problem.			
48	Friday, 30 November 2007	Insertion Sort Expressing sorting algorithms in code. Key technique: Insertion. Analyzing insertion sort.	Sorting	Insertion Sort	Project
Week 13: Project					
49	Monday, 3 December 2007	Merge Sort More efficient sorting techniques. Divide and conquer, revisited. Merge sort. Analyzing merge sort.	Merge Sort	Merge Sort	
50	Tuesday, 4 December 2007	Project Assessment (1) External assessment of images.			

51	Thursday, 6 December 2007	Project Assessment (2) Students discuss programming techniques.			
52	Friday, 7 December 2007	Project Assessment (3) Other explorations of the project. Thoughts on the project?			Exam 3: Types and Algorithms
Week 14: Wrapup					
53	Monday, 10 December 2007	Restricted-Access Collections: Stacks, Queues, and Priority Queues Keeping track of computing tasks. Restricted-access collections: A general structure. Stacks: FIFO collections. Queues: LIFO collections. Priority queues: Prioritized collections.	Keeping Track of Tasks with Restricted Access Collections	Stacks, Queues, and Priority Queues	
54	Tuesday, 11 December 2007	What is Computer Science What is CS? Subfields of CS. Related Disciplines.			
55	Thursday, 13 December 2007	Wrapup The subject matter of the course. Course evaluation. Final thoughts.			
56	Friday, 14 December 2007	Review for the Final Policies for the final. Likely kinds of problems.			Assignment 16: Class Assessment

by John David Stone and Henry Walker and used by permission.) This material is based upon work partially supported by the National Science Foundation under Grant No. CCLI-0633090. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This work is licensed under a Creative Commons Attribution-NonCommercial 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.