

## Course Syllabus

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, 8, break, 9, 10, 11, 12, 13, 14.

	Date	Topic	Reading	Lab	Assignments
<b><i>Week 1: Getting Started</i></b>					
01	Tuesday, 22 January 2008	<b>An Introduction to Algorithms</b> Introduction: What is CS? Exercise: Drawing smileys.		Drawing Smiley Faces	
02	Wednesday, 23 January 2008	<b>An Introduction to CSC151</b> Lessons from day one. Common parts of an algorithm. About the course. Getting started with Linux. Some administrative details.	Grinnell's Linux Environment	Getting Started with Linux	Assignment 1: Class Basics
03	Friday, 25 January 2008	<b>An Introduction to the GIMP</b> What is GIMP? Why GIMP in 151? Short Demo. Lab + Reflection.	The GNU Image Manipulation Program	The GNU Image Manipulation Program	
<b><i>Week 2: Getting Started With Media Computation</i></b>					
04	Monday, 28 January 2008	<b>An Introduction to Scheme</b> Why use programming languages? Scheme basics. Scheme history.	The DrScheme Program-Development Environment & Beginning Scheme & How Scheme Evaluates Expressions (version 1)	Starting Scheme	
05	Tuesday, 29 January 2008	<b>Tools for Scripting the Gimp</b> A model of images, extended. Coding algorithms for drawing. Other useful techniques.	GIMP Tools	Lab: GIMP Tools	

06	Wednesday, 30 January 2008	<b>Turtle Graphics</b> Modeling images through process: Turtle graphics. Some historical notes. Turtle graphics in DrFu.	Turtle Graphics	Turtle Graphics	Assignment 2: Drawing Smileys, Revisited
07	Friday, 1 February 2008	<b>Drawings as Values</b> Thinking about drawings through composition/decomposition. Pure approaches vs impure approaches.	Drawings as Values	Drawings as Values	Quiz 1
<b>Week 3: Scheme Basics</b>					
08	Monday, 4 February 2008	<b>Writing Your Own Procedures</b> Why define your own procedures? How to define your own procedures.	Writing Your Own Procedures & How Scheme Evaluates Expressions (version 2)	Writing Your Own Procedures	
09	Tuesday, 5 February 2008	<b>Computing with Symbols and Numbers</b> Types. Kinds of Numbers. Key Numeric Operations.	Numeric Values & Symbolic Values	Numeric Computation	
10	Wednesday, 6 February 2008	<b>Representing Images as Lists of Spots</b> Lists in Scheme. Basic list operations. Spot lists: Another perspective on images.	Representing Images as Lists of Spots	Representing Images as Lists of Spots	Assignment 3: Algorithmically Creating Simple Images
11	Friday, 8 February 2008	<b>Spot Lists, Continued</b> Quick review. Lab, continued.	Representing Images as Lists of Spots	Representing Images as Lists of Spots	Quiz 2
<b>Week 4: Raster Graphics</b>					
12	Monday, 11 February 2008	<b>Raster Graphics and RGB Colors</b> Representing images, revisited. Pixels and colors: The basics. RGB colors.	Raster Graphics: Images from Pixels and Colors and RGB Colors	Raster Graphics and RGB Colors	

13	Tuesday, 12 February 2008	<b>Documenting Programs and Procedures</b> The need for documentation. The Six P's - a strategy for documenting procedures. Practice.	Documenting Your Procedures		
14	Wednesday, 13 February 2008	<b>A Design Perspective</b> Approaching colors. Managing the huge palette. Other design issues.			Exam 1: Scheme Basics
15	Friday, 15 February 2008	<b>Discussion of Exam 1</b> General issues. Issues with particular problems. Additional Q&A			Quiz 3
<b>Week 5: Control</b>					
16	Monday, 18 February 2008	<b>Boolean Values and Predicate Procedures</b> 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	
17	Tuesday, 19 February 2008	<b>Conditionals</b> Choosing between two options with <code>if</code> . Making multiple choices with <code>cond</code> .	Conditionals	Conditionals	
18	Wednesday, 20 February 2008	<b>Iterating Over Lists</b> Review: How Scheme evaluates expressions. Repetition. Building new lists from old with <code>map</code> . Anonymous procedures. Doing something with each value in a list with <code>foreach!</code> . Drawing lists of spots.	Iterating Over Lists	Iterating Over Lists	Assignment 4: Blending Colors
19	Friday, 22 February 2008	<b>Naming Local Values</b> Why name things. Naming things with <code>let</code> . Naming things with <code>let*</code> . Naming procedures. Lab.	Local Bindings	Local Bindings	Quiz 4

<b>Week 6: Transformations</b>					
20	Monday, 25 February 2008	<b>Transforming Colors</b> Computing new colors from old. Some basic transformations. Transforming pixels and images.	Transforming RGB Colors	Transforming RGB Colors	
21	Tuesday, 26 February 2008	<b>Transforming Images</b> Review: Transforming colors. Transforming images by transforming each pixel. Sequencing transformations with compose.	Transforming Images	Transforming Images	
22	Wednesday, 27 February 2008	<b>Iterating Over Positions</b> Iterating over positions. Blends and other positionally-computed images. Drawing simple shapes.	Building Images by Iterating Over Positions	Building Images by Iterating Over Positions	Assignment 5: List Iteration
23	Friday, 29 February 2008	<b>Recursion Basics</b> The idea of recursion. A sample recursive procedure: sum. Another example: Filtering.	Recursion Basics	Recursion Basics	Quiz 5
<b>Week 7: Recursion</b>					
24	Monday, 3 March 2008	<b>Recursion Basics, Continued</b> Designing a recursive procedure. More time for lab.	Recursion Basics	Recursion Basics	
25	Tuesday, 4 March 2008	<b>Recursion with Helper Procedures</b> Delayed evaluation in recursive procedures. A strategy: Carry along intermediate results. Using recursive helpers. A term: Tail recursion. Designing recursive procedures.	Recursion with Helper Procedures	Recursion with Helper Procedures	

26	Wednesday, 5 March 2008	<b>Other Forms of List Recursion</b> Questions and answers. Patterns of list recursion. Recursive predicates.	List Recursion, Revisited	List Recursion, Revisited	Assignment 6: Color-Reducing Filters
27	Friday, 7 March 2008	<b>Preconditions, Revisited</b> Verifying preconditions. The throw procedure. Husk and Kernel programming.	Verifying Preconditions	Verifying Preconditions	Quiz 6
<b>Week 8: Further Topics</b>					
28	Monday, 10 March 2008	<b>Naming Local Procedures</b> Why have local procedures. Creating local procedures with <code>letrec</code> . Creating local procedures with named <code>let</code> . An example: <code>reverse</code> .	Local Procedure Bindings	Local Procedure Bindings	
29	Tuesday, 11 March 2008	<b>Numeric Recursion</b> Recursion, Generalized. Thinking About Natural Numbers. Numeric Recursion.	Numeric Recursion	Numeric Recursion	
30	Wednesday, 12 March 2008	<b>Geometric Art Through Numeric Recursion</b> Parallel lines, Concentric Circles, and Beyond.	Geometric Art	Geometric Art	Exam 2: Control: Conditionals, Iteration, and Recursion
31	Friday, 14 March 2008	<b>Class Cancelled</b> SamR goes to a conference. Students get a pre-break break.			
<b>Spring Break!</b>					
<b>Week 9: Files</b>					
32	Monday, 31 March 2008	<b>Characters and Strings</b> Representing text. Characters: The basic building blocks. Combining characters into strings.	Characters and Strings	Characters and Strings	

33	Tuesday, 1 April 2008	<b>File Basics</b> File basics: Data persistence and beyond. Ports. Basic file operations.	Files	Files	
34	Wednesday, 2 April 2008	<b>Storing Images as Simple Pixel Maps</b> Files, revisited. Storing images. Storing colors. Storing images, revisited.	Pixel Maps: A Technique for Storing Images	Pixel Maps	Assignment 7: Spirograph-Like Drawings
35	Friday, 4 April 2008	<b>More Efficient Pixel Maps</b> Review. Storing data more efficiently. Tradeoffs.	Pixmap, Revisited: Encoding Data More Efficiently	Pixmap, Revisited	Quiz 7
<b>Week 10: Miscellaneous</b>					
36	Monday, 7 April 2008	<b>More Efficient Pixel Maps, Revisited</b> Quick review. More time for lab.	Pixmap, Revisited: Encoding Data More Efficiently	Pixmap, Revisited	
37	Tuesday, 8 April 2008	<b>Color Palettes</b> Saving space by approximating colors. One strategy: Web-safe colors. Another strategy: Color palettes. Choosing palette colors. Using different palettes.	Color Palettes	Color Palettes	
38	Wednesday, 9 April 2008	<b>Representing Color Palettes with Vectors</b> Problems with lists. A solution: Vectors. Important vector procedures. Representing palettes with vectors.	Vectors	Vectors	Assignment 8: Run-Length Encoding
39	Friday, 11 April 2008	<b>Analyzing Procedures</b> Comparing algorithms. Two related metrics: Time and Number of steps. Counting procedure calls by printing. Tools for analysis.	Analyzing Procedures	Analyzing Procedures	Quiz 8
<b>Week 11: Binary Structures</b>					

40	Monday, 14 April 2008	<b>Pairs and Pair Structures</b> Memory and naming. Pairs and cons cells. Why care?	Pairs and Pair Structures	Pairs and Pair Structures	
41	Tuesday, 15 April 2008	<b>Deep Recursion</b> Lists, revisited. Trees, introduced. Deep recursion, considered. Playing with color trees.	Deep Recursion	Deep Recursion	
42	Wednesday, 16 April 2008	<b>Association Lists and Searching</b> Storing information in tables. Representing table entries as lists. Representing tables as lists. Association lists: Scheme's standard table representation. Implementing key association list procedures.	Association Lists	Association Lists	Assignment 9: Palettes and Vectors
43	Friday, 18 April 2008	<b>Project Ideas</b> More about the project. Technique one: Color Trees. Technique two: Fractals. Technique three: Genetic Art.	Project Ideas	Playing with Project Ideas	Quiz 9
<b>Week 12: Sorting</b>					
44	Monday, 21 April 2008	<b>Higher Order Procedures</b> Elegance. Procedures as parameters. Procedures as return values. Writing map. Writing all?.	Higher-Order Procedures	Higher-Order Procedures	
45	Tuesday, 22 April 2008	<b>Binary Search</b> The problem of searching. Analyzing algorithmic efficiency. Making searching more efficient using divide-and-conquer.	Searching	Binary Search	

46	Wednesday, 23 April 2008	<b>Introduction to Sorting</b> The problem of sorting. Writing sorting algorithms. Examples: Insertion, selection, etc. Formalizing the problem.			
47	Friday, 25 April 2008	<b>Insertion Sort</b> Expressing sorting algorithms in code. Key technique: Insertion. Analyzing insertion sort.	Sorting	Insertion Sort	Quiz 10
<b>Week 13: Project and Beyond</b>					
48	Monday, 28 April 2008	<b>Merge Sort</b> More efficient sorting techniques. Divide and conquer, revisited. Merge sort. Analyzing merge sort.	Merge Sort	Merge Sort	
49	Tuesday, 29 April 2008	<b>Project Assessment: Images</b> External assessment of images.			
50	Wednesday, 30 April 2008	<b>Project Assessment: Algorithms</b> Students discuss programming techniques.			
51	Friday, 2 May 2008	<b>Randomized (Unpredictable) Drawings</b> Random art. Why use randomness. The random procedure.	Randomized Drawing	Randomized Drawing	Exam 3: Types and Algorithms
<b>Week 14: Wrapup</b>					
52	Monday, 5 May 2008	<b>What is Computer Science? Revisited</b> What is CS? Subfields of CS. Related Disciplines.			
53	Tuesday, 6 May 2008	<b>Pause for Breath</b> Pause. for. breath.			

54	Wednesday, 7 May 2008	<b>Wrapup</b> The subject matter of the course. Course evaluation.			Course Assessment
55	Friday, 9 May 2008	<b>Review for the Final</b> Policies for the final. Likely kinds of problems. Questions on Exam 3. Debriefing. Final comments.			

Copyright © 2007-8 Janet Davis, Matthew Kluber, and Samuel A. Rebelsky. (Selected materials copyright 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.