
CS3: **Introduction to Symbolic Programming**

Lecture 13: Final Projects

Spring 2008

Nate Titterton
nate@berkeley.edu

Schedule

12	Apr 14-18	Lecture: Lists Lab: Generalized lists and trees Sequential Programming
13	April 21-25	Lecture: Introduction to the big project Advanced lists Lab: Big project – introduction, and choose partners (checkoff #1)
14	April 28 – May 2	Lecture: Advanced lists Scheme vs. other Languages Lab: Big project (checkoff #2)
15	May 5 – 9	Lecture: (guest) CS at Berkeley and outside... Lab: Big project (checkoff #3 and due at end)
16	May 12	Final Exam Review Lab: <i>no thank you!</i>

The Big Project

- **Two possible projects:**
 - **Connect4**
 - **Blocks World**
- **You can, and should, work in partnerships**
- **You will have three weeks to work on this (it is due on the last lab)**
- **Worth 15% of your final grade**

Project Check-offs

- **There are 3 checkoffs**
 - You need to do them on time in order to get credit for the project**
- 1. **Tell your TA which project you will do and who you will do it with**
- 2. **Show your TA that you have accomplished something. S/he will comment.**
- 3. **Show that you have most of the work done: your TA will run your code.**

Due dates on the final project

Tues/Wed	Thur/Fri
<i>(April 22/23)</i> Introduction	<i>(April 24/25)</i> Checkoff 1
<i>(April 29/30)</i>	<i>(May 1/2)</i> Checkoff 2
<i>(May 6/7)</i> Checkoff 3	<i>(May 9th, Friday)</i> Due (at midnight)

Lets see the projects in action

What issues matter

- **Does it work?**
 - **This is a primary grading standard...**
- **Programming style**
- **Data abstractions**
- **Reading specifications carefully**
- **Adequate testing**

Working in partnerships

- **Highly recommended!**
 - For those of you continuing with CS, you'll be doing this for many future projects
- **Won't be faster, necessarily**
 - While you are less likely to get stuck, there will be a lot of communication necessary
- **A big benefit will be with testing**
- **Remember, only one grade is given...**
 - this grade will be the same, whether the project is a solo or a partnership

Data structures

- **The format of data used in these projects in a central feature**
 - A "data structure" (abstract data type) is a specification of that format. Here, generally, lists of lists (of lists).
 - Accessors and constructor allow for *modularity*: letting parts of a program work independently from other parts.

Sequential Programming

- **Both projects use sequential programming.**
 - drawing graphics,
 - printing, and
 - user input.

 - You won't need to change these procedures, but understanding them will be helpful. And, it may be fun to make simple changes
- **You'll use random in both projects**