

---

# **CS3:** **Introduction to Symbolic Programming**

Lecture 13:  
Final Projects  
Lists (and trees)

**Fall 2007**

**Nate Titterton**  
**nate@berkeley.edu**

# Schedule

---

12	Nov 12–16	Lecture: <i>Holiday</i> Lab: <b>Advanced Lists, Sequential Programming</b> Find partners for the Big Project
13	Nov 19–23	Lecture: <b>Introduction to the Big Project</b> <b>Advanced Lists</b> Lab: <b>Work on the Big Project: checkoff #1</b>
14	Nov 26–30	Lecture: <b>Advanced Lists</b> <b>Scheme versus other languages</b> Lab: <b>Big Project: checkoff #2</b>
15	Dec 3–7	Lecture: <b>CS at Berkeley and outside..</b> Lab: <b>Big Project: checkoff #3 and due</b>
16	Dec 10	Lecture: <b>Exam Review</b> Labs: <i>No thank you</i>

---

## **Any questions about Midterm #2?**

- 1) Replace**
- 2) Word-swap**
- 3) Election analyses**
- 4) Longest-run**
- 5) Maze (lead-to-exit?)**

# 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**
- 3. Tell your TA which project you will do and who you will do it with**
- 4. Show your TA that you have accomplished something. S/he will comment.**
- 5. Show that you have most of the work done: your TA will run your code.**

# Project Check-offs

---

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

# Due dates on the final project

---

<b>Tues/Wed</b>	<b>Thur/Fri</b>
<i>(Nov 20/21)</i> <b>Introduction Checkoff #1</b>	<i>(Nov 22/23)</i> <b><i>Thanksgiving</i></b>
<i>(Nov 27/28)</i> <b>Checkoff #2</b>	<i>(Nov 29/30)</i>
<i>(Dec 4/5)</i> <b>Checkoff #3</b>	<i>(Dec 7<sup>th</sup>, Friday)</i> <b>Due (at midnight)</b>

---

**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.

# Functional 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.**