

**College of the Holy Cross**  
**CCSCNE '06 Programming Contest**  
Schedule, Directions, and System Information

### Schedule Summary

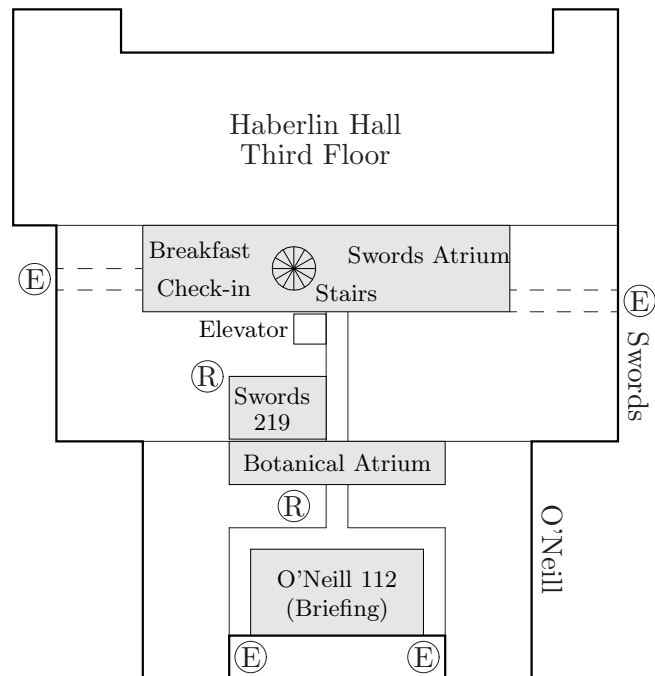
The CCSCNE '06 Programming Contest will be held at the College of the Holy Cross on Friday, April 21, 2006, from 9 AM–noon, in Swords 219.

- **7:30–8:20 AM** Check-in and registration (lower end of the Swords Atrium)
- **7:30 AM** Continental breakfast (near check-in)
- **7:30–8:20 AM** Login and system practice (Swords 219)
- **8:20–8:50 AM** Pre-contest briefing (O’Neill 112)
- **9:00 AM–Noon** Contest (Swords 219)
- **Noon–12:45 PM** Lunch (Swords Atrium)

### Finding the Science Complex

The science complex adjoins College Street, near Gates 4 and 5. Public parking is available near the Hogan Campus Center; turn into the campus at Gate 7 (Campus Map #42) and follow the road to the top of the hill. Public parking is in the lots uphill on your right. Several foot paths lead to the science complex; just head downhill and toward College Street. You’ll enter O’Neill on the first or third floor, Swords hall on the first floor, or Haberlin on the second floor, depending which path you choose.

Two features of the science complex tend to confuse visitors. First, the buildings lie on a hill, so the first floor of O’Neill, the second floor of Swords, and the third floor of Haberlin are contiguous. Second, there is no passage between O’Neill and Swords except on the first floor of O’Neill/second floor of Swords (shown).



- ⓔ Entrance (not all are shown)
- Ⓜ Restroom

To get between floors of Swords, use the spiral stairs in the atrium, or the adjoining elevator.

## Contest Information

A continental breakfast will be available in the Swords atrium at 7:30. **The contest lab will be open by 7:30. The pre-contest briefing will run from 8:20 to 8:50** in O'Neill 112. Lunch will be provided at noon.

Team problem-solving will be done in rooms on the third floor of Swords hall, but the lab is only large enough to accommodate one person per team at a time. You must obtain your team's pass from the lab monitor when you enter the lab, and should return the pass when you leave. The monitor will not allow you into the lab without your team's pass.

The contest machines are Sunray terminals that boot from a remote file server running Red Hat Enterprise Server. Programs may be written in C or C++ (gcc-3.2.3), or Java (JRE 1.5.0). Available editors are `emacs`, `nano`, and `vim`. The contest machine is not connected to the Internet. If you have questions, particularly after the practice run, please ask at the pre-contest briefing.

## Judging Software

Contest submissions and judging are handled by PC<sup>2</sup>. Your team information packet contains team documentation, which is also available on the web at Cal State Sacramento's PC<sup>2</sup> site: <http://www.ecs.csus.edu/pc2>

## Logging In

Locate the workstation in Swords 219 that has your team number on it; your team will use this machine throughout the contest. If the screen is blank, press a key or move the mouse to get a login prompt. Type your login name and password into the appropriate boxes, then click `Login` to log onto the system. Your account name and password are contained in your team packet; the account name has the form `teamxx`, with `xx` your 1- or 2-digit team number.

When you log in successfully, a graphical desktop environment will start. If a terminal (command window) does not open automatically, click on the application launcher near the bottom of the screen, highlight the drop-down menu for Utilities, and select Terminal.

You must start a "PC<sup>2</sup> client" in order to submit solutions: The PC<sup>2</sup> client allows you to interact with the judging software. Once the contest starts, it's best not to exit the PC<sup>2</sup> client. To start a PC<sup>2</sup> client, type the command "`pc2team`" (without the quotes) in a terminal window. Notes:

- Be sure to type the command in lowercase; the contest environment is case-sensitive.
- You must run the command `pc2team` from your home directory; don't change directories before issuing the command.
- You may run the process in the background, but system messages may be printed in the window from which `pc2team` is run. (Please disregard if you don't know what this means. 😊)

A PC<sup>2</sup> login window will open, with blanks for your name and password. These are the same as your login account name and password. Enter them into the blanks and press `Logon`. Your team packet contains the PC<sup>2</sup> contestant's guide; please run through the guide's instructions during the practice run, and notify one of the contest officials if you

encounter problems. Make sure your team’s coder(s) is/are able to perform the following tasks in PC<sup>2</sup>:

- Display the PC<sup>2</sup> Main Menu,
- Select a problem and language,
- Perform test runs, and submit solutions,
- Receive responses from the judges.

## Command Environment

The default shell, `bash`, features tab completion, command editing, and history searching. You should not need to type many commands during the contest, but the following notes may be helpful:

- You may type part of a command and press TAB (the tab key) to see a list of available completions.
- You may use the arrow and backspace keys, and/or `emacs` editing commands, to position the cursor and edit a command without erasing the end of the line.
- The up- and down-arrow keys scroll you through the command history—the list of commands you’ve typed so far. `emacs` search commands can be used to navigate the command history.

## Text Editors

Programs must be written in a text editor: `emacs`, `nano`, or `vim`. You are assumed to be familiar with at least one of these. Your team packet contains command reference sheets for `emacs` and `vi`. These reference sheets come from Rice University, and have not been exhaustively inspected for perfect compatibility. However, basic editor commands work as advertised.

To create a file named `problem1.cc` with `emacs`, type “`emacs problem1.cc &`” at a prompt. (The “`&`” allows you to type subsequent commands in the same window.) Enter your code into the editor window. To save your work, type `C-x C-s` (control-x control-s) or select Save (current buffer) from the File drop-down menu in `emacs`. Once you have saved your program, you can use PC<sup>2</sup> to test and submit it.

## Compiling and Testing

The PC<sup>2</sup> client allows you to test your programs for compile errors, and to inspect their output, before you submit the program for judging. Please consult the contestant’s guide for details.

If you prefer, you may test your programs manually. The table below summarizes the commands, assuming your input file is named `myprog` followed by an appropriate extension:

Language	To compile	To execute
C++	<code>g++ myprog.cc</code>	<code>./a.out</code>
C	<code>gcc myprog.c</code>	<code>./a.out</code>
Java	<code>javac myprog.java</code>	<code>java myprog</code>

PC<sup>2</sup> executes the same commands, so there is no advantage to compiling manually.