

# CSC207H // Software Design // Fall 2011

**Course objectives:** Welcome to CSC207H, an introduction to software design. One major goal of this course is to introduce you to large-scale software design and development concepts and tools that become useful as you work on projects in teams. We will discuss effective team behaviors and communication skills, practice agile methods for designing software, and use tools such as a fully-featured IDE and a version control system.

The other primary goal of this course is to help you practice how to learn a new language. Instead of focusing on syntax as we did in CSC108H, we will compare salient features of Python and Java, expecting you to fill in details outside of class, and will investigate Java's memory model, scoping facilities, and object-oriented structures in depth. By the end of this course, we believe that you will pick up new languages and tools with little guidance and expect that you will be prepared to develop code in small teams.

|              |   |                       |
|--------------|---|-----------------------|
| Instructor   | Paul Gries  |                       |
| Office       | BA 4234   |                       |
| Office Hours | T 2-3 (shared with CSC148H), W 2:15-3:30  |                       |
| Email        | <a href="mailto:pgries@cdf.toronto.edu">pgries@cdf.toronto.edu</a> (please begin email subjects with "207") |                       |
| Phone        | 416-978-6322  |                       |
| Lectures     | L0101: MW 12 in BA1190  | L5101: W6-8 in BA1180 |
| Labs         | Start the second week of term   |                       |
| Website      | <a href="http://www.cdf.toronto.edu/~csc207h/fall/stg">http://www.cdf.toronto.edu/~csc207h/fall/stg</a>     |                       |

**Required background:** All the material from CSC148H: object-oriented programming, a working knowledge of basic data structures (stacks, queues, trees, linked lists), recursion, searching, and sorting.

**Required reading:** The course website will contain (sketchy) lecture notes, the policy on missed work, a link to the discussion board, and more. All announcements made in lecture and on the discussion board are required reading. Code examples will be shared via a *repository* (see week 1 version control lecture slides). There is no required textbook for the course; any required readings will be posted in electronic format to the course website. However, if you wish to own a hardcopy Java reference, you may find *Developing Java Software, third edition* by R. Winder and G. Roberts to be helpful.

**Contacting the instructor:** Please use email for personal issues and the discussion board to ask general course-related questions. I receive a large quantity of email over the term but try to respond by the end of the next day. However, it may take longer, especially on weekends and near due dates. Due to high volume, email sent close to a due date may not get a timely response, so get started on any work as early as possible in case you have questions.

Begin email subject lines with "207" lest your message accidentally be filed as spam.

**CDF account and email forwarding:** You have a CDF account that you will use to submit your work and to do the labs. We will occasionally send crucial course announcements to that account. You must check that email frequently. Probably the best way to do this is to forward email to your UTORmail address, since the university expects you to read it anyway. Here is how to set up email forwarding: <http://www.cdf.toronto.edu/cdf/faq.html#MAIL4>

**Marking scheme:** A0 (5%), A1 (10%), A2 (10%), A3 (10%), Teamwork (5%), Lab participation (5%), 3 Quizzes (10% each, best 2 of 3), Final Exam (35%)

**Midterm and final exam:** The course has three quizzes that will take place at the beginning of three Wednesday lectures; lecture will continue after the quiz. There is a three-hour final exam. The final exam is comprehensive, and you must obtain a mark of at least 40% to pass the course; otherwise, a grade of no higher than 47% will be assigned.

**Assignments:** There are four assignments. The first and last are to be done *individually*. The others involve group work. Assignments are to be submitted via the Subversion version control system. Files *committed* to your repository will be accessible to you, the TAs, and myself. You may *commit* changes as often as you like; we will mark the last *revision* submitted before the deadline.

Do not worry if you do not understand the italicized words – we will be discussing Subversion (and version control systems in general) during the first week of class!

**Tutorials/Labs:** Lab attendance is mandatory and you will be assigned small pieces of homework from time to time to bring to lab. A few of your assignments will be done in groups involving others in your tutorial section; you will have a chance to apply what we have covered in class on how to work effectively as a team. During assignment-focused labs, you will be meeting with the TA assigned to be your "scrum master" for your assignments (more about this in class) and your team members.

**Late policy:** There are no grace days; **all due dates are firm**. In case of illness, please have a doctor complete an official U of T medical certificate. For other emergencies, be prepared for us to request some kind of documentation.

**Academic Offences:** A0, A3, the quizzes, and the final exam must be done by you *alone*, and your work must not be submitted by anyone else. The following is required reading:

<http://www.utoronto.ca/academicintegrity/academicoffenses.html>

The department uses software that compares programs for evidence of similar code, and we search the web for posted solutions. Please read the Rules and Regulations from the U of T Calendar (especially the Code of Behaviour on Academic Matters):

<http://www.artsandscience.utoronto.ca/ofr/calendar/rules.htm>

Please don't cheat. It is unpleasant for everyone involved, including us. Here are a couple of general guidelines to help you avoid breaking the rules:

- Never look at another person's solution (except for the team assignments), whether it is on paper or on the computer screen, and never show another student your solution. This applies to all drafts of exercise and assignment solutions and to incomplete solutions, including pseudocode and diagrams.
- The easiest way to avoid committing an offence is to only discuss the piece of work with your assignment partner(s), the course TAs, the CDF Help Centre TAs, and Paul.

| Week | M-F dates      | Coursework                             | Reminders  |
|------|----------------|--|--|
| 1    | 12–16 Sep      |  | First week of classes! Wahoo!                                      |
| 2    | 19–23 Sep      |  | Sun 25 Sep: last day to add courses<br>Labs start                  |
| 3    | 26–30 Sep      | A0 due Thursday 29 Sep 10:00pm         |  |
| 4    | 3–7 Oct        | Quiz 1 (in Wed lecture)                |  |
| 5    | 10–14 Oct      | A1 due Thursday 13 Oct 10:00pm         | Mon 10 Oct: Thanksgiving<br>Fri 14 Oct: exam timetable posted      |
| 6    | 17–21 Oct      | Quiz 2 (in Wed lecture)                |  |
| 7    | 24–28 Oct      | A2 part I due Thursday 27 Oct 10:00pm  | Study for all your midterms  |
| 8    | 31 Oct – 4 Nov |  | Thu 3 Nov: last day to drop classes                                |
| 9    | 7–11 Nov       | A2 part II due Thursday 10 Nov 10:00pm | Mon, Tue 7–8 Nov: fall break                                       |
| 10   | 14–18 Nov      | Quiz 3 (in Wed lecture)                |  |
| 11   | 21–25 Nov      |  |  |
| 12   | 28 Nov – 2 Dec | A3 due Thursday 1 Dec 10:00pm          |  |
| 13   | 5–9 Dec        |  | Last week of classes! Wahoo!<br>Wed 7 Dec: makeup Monday for L0101 |
| X    | 9–20 Dec       | Final exam                             | Final exam period  |