

Course Outline

## SYSC 5800: Network Computing

Winter 2007

Instructor: Thomas Kunz

[tkunz@sce.carleton.ca](mailto:tkunz@sce.carleton.ca)

Office Hours: Mondays 1-2 pm, Wednesdays 5-6 pm

Design and implementation of distributed applications that use data networks (telecommunications networks/Internet/wireless networks) as their computing platform. The course content covers: Basics of networking; socket programming; Java networking facilities; Introduction to open distributed processing; RPC and RMI; CORBA; WWW, XML, Web Services (and potentially P2P and Grid Computing). In essence, we will cover a range of technologies developed over the past 20+ years to solve the fundamental challenge of distributed computing: locating and accessing remote components. The earlier technologies typically produce tight coupling, resulting in brittle systems: one thing goes wrong and the whole system breaks. This was exemplified by Lamport's famous (or infamous) definition of distributed systems: *A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.* More recent work/technologies have focused on building distributed applications from loosely coupled components that can be dynamically composed (the "state-of-the-art" buzzword for that trend is *Service Oriented Architecture*). The course is designed as a hands-on course, with a heavy emphasis on developing your own distributed applications.

**Resources:** There is no single textbook that adequately covers the diverse range of topics (and buying multiple textbooks is clearly not very appealing financially). However, some somewhat good books to cover parts of the course are listed below:

- *Distributed Systems: Principles and Paradigms*, by Andrew S. Tanenbaum and Maarten van Steen, Prentice Hall 2002, ISBN 0-13-088893-1.
- *Java Network Programming*, 2nd edition, by Merlin Hughes et al., Manning Publications 1999, ISBN 1-884777-49-X.
- *Building Web Services with Java*, 2<sup>nd</sup> edition, by Steve Graham et al., Sams Publishing 2005, ISBN 0-672-3264.
- *An Introduction to XML and Web Technologies*, by Andres Møller and Michael Schwartzbach, Addison-Wesley 2006, ISBN 0-321-26966-7.

I will post my slides, plus other reference material, on the course webpage, <http://kunz-pc.sce.carleton.ca/sysc5800/>. I will also post marks, etc. on this webpage.

**Marking Scheme:** There will be a final exam worth 30%, two in-class presentations worth 10% each, and a course project, which is worth 50% total (10% will be assigned based on the project proposal, and 40% will be based on the final project itself). The final exam is for evaluation purposes only and will **not** be returned to students. All documents (project proposal, final report) have to be submitted as a hardcopy and softcopy, and follow certain formatting guidelines (in

**Carleton University**  
**Department of Systems and Computer Engineering**

**SYSC 5800**

**Network Computing**

**Winter 2007**

**Course Outline**

particular length and font size limitations). See below for a discussion of these requirements. All coursework, with the obvious exception of the final exam, will be based on group efforts. Group sizes are expected to be approximately 3-4, depending on course enrollment, and groups will be finalized after the end of the registration deadline, January 19. You can either form teams of your own (starting immediately) or you will be assigned to a group by the instructor by January 22.

**Due Dates:** The final exam will be in class on Monday, April 3. Team formations will be finalized by January 23. The project proposal is due in class on Monday, February 26. The first round of in-class presentations is the week before Reading Week (i.e., on February 12 and 14). The second round is starting March 12, with the presentation order determined by a lottery on March 12. The hardcopy of the final project report is due April 9, in my office, by noon, the softcopy (submitted via e-mail to [tkunz@sce.carleton.ca](mailto:tkunz@sce.carleton.ca)) is due the same time.

	Due Date	Weight
Team Formation and Topic Assignment	January 22	
Project Proposal	February 26	10 %
First round of presentations (PPT files due Feb. 11)	February 12/14	10%
Second round of presentations (Material due March 11)	Starts March 12	10 %
Final Exam	April 2	30 %
Final Project Report	April 9	40 %

**Cheating:** Suspected academic offences (cheating) will be investigated and reported to the Dean of Graduate Studies and Research, see also General Regulation 14. Cheating covers a wide range of offences, including submitting another student's work as your own solution, or committing an act of plagiarism. A more detailed description of what constitutes plagiarism is available on the WWW: (<http://www.zoology.ubc.ca/bpg/Advising/Plagiarism.htm>).

**Project:** 50% of your mark will be determined by a course project (10% for the proposal, 40% for the final project). The overall goal is to apply the course lessons in a programming project based on Jabber ([www.jabber.org](http://www.jabber.org)), on open-source instant messaging platform. Possible topics are enhancing the core Jabber functionality, implementing an interesting application on top of Jabber, providing new/additional gateways to other systems, porting Jabber clients/servers to new platforms, etc. The project proposal should identify the topic and discuss related work (why is the topic relevant/interesting, has similar functionality/service been implemented in any other system, what expectations do you have for the final project outcome). The proposal should not exceed 5 single-spaced pages (including references, etc.). I will provide feedback on the proposals by e-mail within a week. The final project should consist of running and well-documented code, an installation manual, a user manual, and a separate document (not to exceed 15 pages) that extends the proposal document: what problem was solved, what related work was reviewed, what are the contributions of the project (functionality, performance), and what are the shortcomings/possibilities for future work. All submissions should be both as hardcopy (drop them off either in my office or my departmental mailbox) **and** as softcopy (submitted to my e-mail address [tkunz@sce.carleton.ca](mailto:tkunz@sce.carleton.ca)). Marks will be based on the completeness of the submission, the innovative aspect of the application, and the quality of the code and

---

**Course Outline**

documentation. Equal value will be given to the code (including installation and user manuals) and the report.

**Presentations:** each group will prepare two presentations. The first set of presentations will be a brief (15 minutes) presentation on a Jabber-related topic, based on a list I will provide. An additional 5 minutes are scheduled for Q&A. This is about the length of time you would be given to present at an academic conference as well. I will arbitrarily assign topics to teams, once they are formed (i.e., by January 22), to ensure a proper order/logic of presentations. The goal of the first presentation is to introduce an aspect of the Jabber system in the form of a tutorial. You need to submit a PPT file electronically to [tkunz@sce.carleton.ca](mailto:tkunz@sce.carleton.ca) by February 11 that will be made available to all students on the course webpage.

In the second round of presentations, each group will give a longer (25 minutes plus 5 minutes for Q&A) presentation on their course project: what are the goals, what are possible design alternatives, progress to-date, outstanding problems. You may choose to use PPT for this presentation or run a demonstration, etc. as you please, with all material you'd like to have access to in the electronic classroom submitted electronically to [tkunz@sce.carleton.ca](mailto:tkunz@sce.carleton.ca) by March 11. The order of presentation will be determined by lottery on March 12 (i.e., you should be ready to present on that day). I will evaluate the presentations and provide feedback to each group after each round.

**Outline:**

- Network Computing Introduction
- Intro to Network Architectures
- Basic Programming Models: Messaging vs. Distributed Shared Memory
- Socket Programming
- Remote Procedure Calls
- Remote Method Invocation
- Common Object Request Broker Architecture
- World Wide Web
- Extensible Markup Language
- Web Services
- P2P
- Grid Computing

**Students with Disabilities:**

Students with disabilities requiring academic accommodations in this course are encouraged to contact a coordinator at the Paul Menton Centre for Students with Disabilities to complete the necessary letters of accommodation. After registering with the PMC, make an appointment to meet and discuss your needs with me. This is necessary in order to ensure sufficient time to make the necessary arrangements. Please note the following deadlines for submitting completed forms to the Paul Menton Centre: **March 10th, 2007** for the Winter Term.