Department of Electrical and Computer Engineering

COURSE OUTLINE

SEng 462: Distributed Systems and the Internet

Term - SPRING 2015 (201501)

Instructor

Dr. Stephen W. Neville
Phone: n/a
E-mail: sneville@ece.uvic.ca

Office Hours

Days: Wed.
Time: 2:30 – 3:30 (or by appointment)
Location: EOW 307 or ELW A228

Note: All course emails ***MUST*** have "SEng462:" in the subject line and ***MUST*** be sent from UVic email accounts. Emails without proper subject lines or sent from off-campus email accounts will likely be dropped by UVic's email spam filters or be automatically redirected to junk email folders.

Lectures

A-Section(s): A01 / CRN 22705 and A02 / CRN 22706
Days: Tues., Wed., Fri.
Time: 12:30-1:20pm
Location: ECS 130

Tutorial

Days: Wed.
Time: 2:30-3:20pm
Location: ELW B208

Full details of all official course locations and times are available from UVic's Timetable web page (http://uvic.ca/timetable). In the case of any discrepancies between the above denoted times and places and the official UVic timetable web page, the official UVic web page is authoritative.

Required Text:

n/a

Optional Text:

Title: Distributed Systems:
Concepts and Design (3rd Ed.)
Author: G. Coulouris, J.Dollimore, and
T. Kindberg
Publisher: Addison Wesley
Year: 2001

Course Web Site: http://www.ece.uvic.ca/~sneville/Teaching/SEng462

Note: Failure to successfully complete the course project will result in a grade of N being awarded for the course.

Assessment:

Project:	40%
Mid-term	20%
Final	40%

Date: Feb 27th, 2015.

Due Dates for Project Materials:
Project milestones will also need to be met throughout the course. Milestones due dates will be listed on the project web site (accessible as a link from the course web site). ***All*** late milestones will receive a zero grade.

Weekly log-book entries will also be required by every student for all weeks of the course project (submitted on-line via the course project web site). Each weekly log-book entry *must* be validated by *all* other members of the students project group. Each missed, improperly entered, and/or improperly validated log-book entry will receive a -1% deduction from the course project to a maximum total deduction of -5%.

The final grade obtained from the above marking scheme will be based on the following percentage-to-grade point conversion:

Passing Grades	Grade Point Value	Percentage for Instructor Use Only	
A+ A A-	9 8 7	90 – 100 85 – 89 80 – 84	
B+ B B-	6 5 4	77 – 79 73 – 76 70 – 72	
C+ C	3 2	65 – 69 60 – 64	
D	1	50 – 59	
Failing Grades	**Grade Point Value**	**Percentage for Instructor Use Only**	**Description**
E	0	0 - 49	Fail, *Conditional supplemental exam. (For undergraduate courses only)
F	0	0 – 49	Fail, no supplemental.
N	0	0 – 49	Did not write examination, Lab or otherwise complete course requirements by the end of term or session; no supplemental exam.

**Assignment of E grade will be at the discretion of the Course Instructor.*

The rules for supplemental examinations are found on page 80 of the current 2014/15 Undergraduate Calendar.

Term in which E Grade Was Obtained	Application Deadline for Supplemental Exam	Supplemental Exam Date
First term of Winter Session (Sept – Dec)	February 28 in the following term	First week of following May
Second term of Winter Session (Jan – Apr)	June 30 in the following term	First week of following September

Summer Session (May – Aug)	October 31 in the following term	First week of following January

Deferred exams will normally be written at the start of the student's next academic term; i.e., approximately 4 months following the deferral of the exam.

Course Description

1. Course Objectives:

   The objectives of this course are to introduce basics of scalability problems that occur within distributed software systems, particularly as these apply to solutions involving Internet and middleware technologies. Students will gain an understanding of how to assess where and why system bottlenecks occur and how these can be resolved. Students will gain the ability to analyze web site user interaction data and how to transform such data in to knowledge about the computing resource requirements needed to meet desired system performance specification. Students will also be introduced to some of the mathematical models that are used to analyze system performance and scalability concerns, as well as some of the basic issues involved in securing software systems. Through the course project students will gain hands on experience with software scalability issues, as well as software instrumentation and analysis approaches.

2. Learning Outcomes:

   Students successfully completing this course will gain an understanding of:

   - The problems that arise when software systems are scaled up to large numbers of users and/or systems events, i.e. into the many millions of daily transactions.
   - How to perform the system testing required to identify where system bottlenecks are occurring and how to resolve them.
   - Why middleware exists, its various forms, and how it is used to construct distributed software systems.
   - What the ACID properties for transaction processing are and why they are important.
   - The engineering principles that underlie the construction of larger-scale software systems that behave predictably, ensure security, etc., and why these are innately challenging engineering problems.

   From the course project, students will gain practical experience in how to build and debug a larger-scale distributed software systems and why this is fundamentally different than building small-scale software systems intended to service only low numbers of users.

3. Syllabus

   The exact pacing of the syllabus materials will vary in accordance with each class, as such the syllabus solely denotes a provisional pacing which may or may not change during the course delivery.

   Course introduction

   Discussion of course project details

   Introduction to distributed systems

   What and Why

- Underlying issues
- Transparency
- Distributed Architectures

Distributed Software Design
- Design Principals
- Design Mechanisms
- Design Methodology

Distribution and Performance Analysis
- Workload Matrix
- Performance Matrix
- Rules for addressing bottlenecks
- Soccer league example

Middleware
- NOS, Stacks, OO-Middleware, Message-oriented Middleware
- Middleware in multi-tiered distributed applications

Persistent State Architectures
- Enterprise Jave Beans

Web services
- Web service architectures
- Implementing web services

Transaction Processing
- Principals
- Concurrency control
- Distributed transactions

Capacity Planning
- Performance measures & rules
- Availability and Reliability

User Behavioural Modeling
- Customer behaviour model graph (CBMG)
- Client/Server interaction diagram (CSID)

Performance Modeling
- Single Queue Model
- Queuing Network Model
- Basic introduction to queuing theory

Security
- Network Security Approaches
- Cryptography

Timing issues in distributed systems (time permitting)

Statistical modeling issue in distributed systems (time permitting)

Note to Students:

Students who have issues with the conduct of the course should discuss them with the instructor first. If these discussions do not resolve the issue, then students should feel free to contact the ECE Chair by email or the ECE Chair's Secretary eceasst@uvic.ca to set up an appointment.

Accommodation of Religious Observance
See http://web.uvic.ca/calendar2014/GI/GUPo.html

Policy on Inclusivity and Diversity
See http://web.uvic.ca/calendar2014/GI/GUPo.html

Standards of Professional Behaviour

You are advised to read the Faculty of Engineering document Standards for Professional Behaviour at http://www.uvic.ca/engineering/assets/docs/professional-behaviour.pdf which contains important information regarding conduct in courses, labs, and in the general use of facilities.

Cheating, plagiarism and other forms of academic fraud are taken very seriously by both the University and the Department. You should consult

http://web.uvic.ca/calendar2014/FACS/UnIn/UARe/PoAcI.html for the UVic policy on academic integrity.

Plagiarism detection software may be used to aid the instructor and/or TA's in the review and grading of some or all of the work you submit.

In addition other automated means of detecting plagiarism and other forms of unprofessional behavior, including but not limited to plagiarism detection software, security software, etc., may be used to aid the instructor and/or TA's in the review and grading of some or all of the work you submit. Work called out by these processes and/or by human adjudication will receive a zero grade and all instances will be reported to the appropriate administrative bodies for further disciplinary action.

As a portion of the course project mark comes from the placement of each student group's project with respect to the other student groups any attempt to improperly gain an advantage over other student groups, e.g., by gaming the automated assessment system, by unethical behaviours, by copying current or prior project code based, or other means, will result in an immediate loss of at least 50% of the total project marks. Such behaviours will also be reported to the appropriate administrative authorities for further disciplinary action.

Note: All materials used in the course are copyrighted by the authors of the materials, whether course notes, exams, assignments, textbook material, etc. Any and all posting, via web sites, social media sources, etc., or reproduction of these materials is a copyright violation and as such contravenes the standards of professional behavior and will therefore be subject to appropriate measures.