Carleton University

Department of Systems and Computer Engineering

SYSC 4101 Software Validation, Verification and Testing Fall 2011

Course Outline

Instructor: Yvan Labiche 520-2600 ext. 5583
Room 7078 MC labiche@sce.carleton.ca
Office hours: Tuesdays and Thursdays from 13h to 14h.
Otherwise, by appointment.

Course Description and Objectives:

Software Verification and Validation (V&V) are two important activities of any software development. Software validation is about checking that we have built the right system whereas software verification is about checking that we have built the system right. One of the main techniques used for V&V is called software testing, and it implies executing the software using actual inputs.

Some data reported in literature indicate that software testing usually amounts for 30 to 40% of the total software development cost, and that for safety critical software this percentage can go up to 70%. Although a lot of testing is conducted, there are still many defects in released software (software not built right) and many software do not entirely satisfy customer needs (not the right software). A 2002 study performed in the USA by the Department of Commerce's National Institute of Standards and Technology (NIST) indicates that "software bugs, or errors, are so prevalent and so detrimental that they cost the U.S. economy an estimated $59.5 billion annually."

One of the main limits of today's testing activities is that they are often not conducted in a systematic, repeatable way, using clear rationale. For instance, a recent study reported that open source software development projects lack "attention to basic, accepted, and mature testing techniques."

The main purpose of this course is to introduce you to these basic, accepted, systematic, mature testing techniques.

The main goals of this course are:

- to be familiar with the concepts of verification, validation, testing, test model, test criteria, …;
- to understand the benefits and limitations of software testing;
- to understand testing techniques for unit testing, integration testing, and system testing;
- to understand the problem of regression testing;
- to understand problems specific to procedural, object-oriented, distributed, or real-time software;
- to understand the different kinds of testing techniques that exist, their advantages and drawbacks;
- to become more familiar with the general notion of quality assurance.

The main objectives of this course are:

- to apply basic, accepted, mature testing techniques;
- to apply testing techniques for unit, integration, and system testing;
- to apply testing techniques to procedural as well as object-oriented software;
- to decide, in context, which testing technique is more adequate.

Carleton University

Department of Systems and Computer Engineering

SYSC 4101 Software Validation, Verification and Testing Fall 2011

Course Outline

Prerequisites:
The prerequisites for SYSC-4101 are SYSC-3100 and SYSC-4800. Note that SYSC-4800 can be taken concurrently.
Students who have not satisfied the prerequisites for this course must either a) withdraw from the course, or b) fill in a prerequisite waiver from the www.sce.carleton.ca/ughelp, or c) may be deregistered from the course after the last day to register for courses.

Lectures:

When: Tuesdays and Thursdays from 11h30 to 13h.
Where: TBD.

Laboratory Sessions (compulsory):

When: Mondays (every other week, starting Sept. 19th) from 11h30 to 14h30.
Where: TBD

Textbook:
Title: Introduction to Software Testing
Authors: Paul Ammann, Jeff Offutt
Publisher: Cambridge University Press
ISBN: 978-0-521-88038-1
Note that no one text book currently available can cover the whole spectrum of software testing. Course notes are therefore equally important to understand the material. Additionally a bibliography of (possible) reference books/articles that are particularly relevant to this course will be posted on the course web site (WebCT).

References: References for SYSC-4101 will be available on the course website.

Grading Scheme:
There will be:

- three assignments, each worth 5% of the final mark;
- a (closed-book) midterm exam worth 25% (held approximately half-way through the term);
- a (closed-book) final exam worth 30%.
- compulsory lab work worth 30%: each of the 6 labs is worth 5% of the final mark.

To pass the course, an appropriate overall mark (D- or higher) must be obtained and a passing mark (D- or higher) must be obtained at the final exam.

Important dates:

- Assignment 1 will be posted on Sept. 22 and due on Oct. 12.
- Assignment 2 will be posted on Oct. 13 and due on Nov. 2.
- Assignment 3 will be posted on Nov. 3 and due on Nov. 23.
- The (in-class) midterm is scheduled on Oct. 20.
- The final exam will be scheduled during the December exam period.

Students with conflicts: see Academic Accommodation below.

Assignments:
The assignments will lead you through the use of software testing techniques, and are a good starting point when preparing the mid-term and final exams. Portions of the work from each assignment may be reused and refined in subsequent assignments, so students are encouraged not to "write-off" any particular assignment just because of its relatively low weight in the overall grading scheme.
Students are encouraged to discuss issues when working on assignments; however, you are expected to submit your own work for grading. There is a fine line between cooperating with your colleagues (discussing problems and ideas) and copying solutions (plagiarism). Not only is plagiarism an instructional offence (see the Undergraduate Calendar), but doing the assigned work by yourself is by far the best way to prepare for the exams.
Submission: Assignments are due at midnight of the due date and must be submitted on WebCT.
Late assignments will be graded according to the following policy: a 20% penalty per day (i.e., 24 hours).

Final Exam: *Is for evaluation purposes only and will not be returned to the student.*
Students who miss the final exam may be granted permission to write a deferred examination (see the Undergraduate Calendar for regulations on deferred exams).

Academic Accommodation
You may need special arrangements to meet your academic obligations during the term. For an accommodation request the processes are as follows.
Pregnancy obligation: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website
http://www.carleton.ca/equity/accommodation/student_guide.htm
Religious obligation: write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website
http://www.carleton.ca/equity/accommodation/student_guide.htm
Students with disabilities requiring academic accommodations in this course must register with the Paul Menton Centre for Students with Disabilities for a formal evaluation of disability-related needs. Registered PMC students are required to contact the Centre, 613-520-6608, every term to ensure that I receive your Letter of Accommodation, no later than two weeks before the first assignment is due or the first in-class test/midterm requiring accommodations. If you require accommodation for your formally scheduled exam(s) in this course, please submit your request for accommodation to PMC by November 11th 2011 for Fall term (December exams).
You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://carleton.ca/equity/accommodation

Plagiarism:
Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated. Please refer to the section on instructional offenses in the Undergraduate Calendar for additional information.

Health and safety:
http://www.sce.carleton.ca/courses/health-and-safety.pdf

Tentative Outline
The following is a tentative outline of the course; it might change, based on time constraints:

Week 1	Introduction to software validation, verification, and testing
Week 2 Week 3 Week 4	Black-box testing Deriving tests from specification: e.g., Category-Partition testing, Boolean logic testing.
Week 5 Week 6 Week 7	White-box testing Deriving testing from the source code: control flow testing, data flow testing.
Week 8 Week 9	Testing Object-oriented systems Testing a class, testing generalizations, …
Week 10 Week 11	Regression testing selection, prioritization, minimization.
Week 12	Testing real-time systems, distributed systems Challenges and solutions.
Week 13	Other V&V techniques

Miscellaneous:
Attendance:
Being (consistently) late in lectures and/or labs does not show professional behavior. Moreover, those students who miss lectures and/or labs may encounter difficulties as far as their final grade is concerned since (i) a lot of the material introduced during lectures is not necessarily in the transparencies provided on the course web site (this may depend on student participations, student questions, …), and (ii) questions during the final exam will target the understanding of lab activities as well as lab material.
Expectations:
I expect the students to invest a substantial amount of time and energy in reading the textbook and doing the assignments/lab work. Looking at the transparencies I will provide may not be enough to achieve the level of understanding that will be required for the mid-term and final exams.