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 are building the right system whereas software verification is about checking that we are building 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%. Other anecdotal evidence of the importance of software testing is that in large projects, the amount of test code (for instance measured in lines of code) can be double the amount of application code! Although a lot of testing is conducted, there are still many defects in released software (software not being built right—verification issue) and software do not entirely satisfy customer needs (not the right software being built—validation issue).

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 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 standard black-box and white-box testing techniques, their advantages and drawbacks;
- to understand the problem of regression testing;
- to understand problems specific to procedural, object-oriented, distributed, or real-time software;
- to become more familiar with the general notion of quality assurance.

The main objectives of this course are:
- to design test suites according to basic, accepted, mature testing techniques;
- to design test suites for unit, integration, and system testing;
- to apply testing techniques to procedural as well as object-oriented software;
- to select testing techniques that are adequate to a software development context;
- to design testing infrastructure to allow the execution of tests.

Prerequisites: Undergraduate studies in software engineering, computer systems engineering, computer science, or related field. You should be familiar with at least one programming language, and preferably an object-oriented language such as C++ or Java. Knowledge of UML is preferable. You should check prerequisites with the instructor.
Lectures:
When: Tuesdays and Thursdays from 10h05 to 11h25.
Where: UC282.

Contact:
Instructor: Yvan Labiche yvan.labiche@carleton.ca

Office hours:
Yvan Labiche by appointment only ME4462

Textbook:
Title: Introduction to Software Testing (second edition)
Authors: Paul Ammann, Jeff Offutt
Publisher: Cambridge University Press
Note that no single textbook currently available can cover the whole spectrum of software testing, verification and validation. Course notes are therefore equally important to understand the material.

Course web site:
Course material will be available on cuLearn: look for the cuLearn link from Carleton’s web site. Although lecture notes are online in advance of lectures, the material introduced in a series of slides may change without prior notice until it is introduced in class. One introduced in class, (unlikely) changes to lecture material will be disclosed. Additionally a bibliography of (possible) reference books/articles that are particularly relevant to this course may be posted on the course web site.

Important: Student or professor materials created for this course (including presentations and posted notes, labs, case studies, assignments and exams) remain the intellectual property of the author(s). They are intended for personal use and may not be reproduced or redistributed without prior written consent of the author(s).

Students from the University of Ottawa:
You can request to have access to cuLearn: please see http://gradstudents.carleton.ca/forms-policies/.
Grading Scheme:
There will be:
- two (closed-book) midterm exams, each worth 20%;
- a (closed-book) final exam worth 35%.
- a course project worth 25%.

Course project
The project is a semester long, significant endeavour. You can expect to dedicate on average four hours per week on the project.
The project is an individual work. No team-work is allowed.

More information online on the course web site: Make sure to read that ASAP.

The project has two deliverables:
- Project proposal (not marked).
  It is your responsibility to submit a proposal. The proposal is meant to precisely describe, like a contract, what you intend to do and how (milestones, deadlines…) you intend to do that so I can provide you with constructive feedback to ensure you are on the right track and your work will lead to a successful project. The proposal is not a final report but it should provide enough details so I can provide feedback.
- Project report is due on Dec. 8.

Important dates (subject to change):
- The first (in-class) midterm is scheduled on Oct. 5
- The second (in-class) midterm is scheduled on Nov. 16.
- The final exam will take place during the December exam period. The 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).
  Unless otherwise specified, mid-term exams will take place in the classroom.
- Project proposal is due on Sept. 22nd.
- Project report is due on Dec. 8. No extension allowed.
Tentative Outline
The following is a tentative outline of the course; it might change, based on time constraints:

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Introduction to software validation, verification, and testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 2</td>
<td>Testing from a plain language specification (black-box testing): the Category Partition testing technique</td>
</tr>
<tr>
<td>Week 3</td>
<td>Testing from graphs (black-box and white-box testing)</td>
</tr>
<tr>
<td>Week 4</td>
<td>Testing from a state model.</td>
</tr>
<tr>
<td>Week 5</td>
<td>Testing from the source code.</td>
</tr>
<tr>
<td>Week 8</td>
<td>Testing Object-oriented systems</td>
</tr>
<tr>
<td>Week 9</td>
<td>Testing a class, testing generalizations, …</td>
</tr>
<tr>
<td>Week 10</td>
<td>Regression testing:</td>
</tr>
<tr>
<td>Week 11</td>
<td>selection, prioritization, minimization.</td>
</tr>
<tr>
<td>Week 12</td>
<td>Revisiting stubs, drivers and oracles by examples.</td>
</tr>
<tr>
<td>Week 13</td>
<td>Testing real-time systems, distributed systems, testing GUIs Challenges and solutions. Other V&amp;V techniques</td>
</tr>
</tbody>
</table>

Miscellaneous:

Attendance:
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:
Students are expected to invest a substantial amount of time and energy in reading the textbook and doing the assignment/lab work. Looking at the transparencies provided may not be enough to achieve the level of understanding that will be required for the mid-term and final 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/
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/
Students with disabilities requiring academic accommodations: The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your Letter of Accommodation at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (if applicable). Requests made within two weeks will be reviewed on a case-by-case basis. After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website (www.carleton.ca/pmc) for the deadline to request accommodations for the formally-scheduled exam (if applicable).
You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at http://www.carleton.ca/equity/.

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: