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:
The prerequisites for SYSC-4101 can be found in the undergraduate calendar: http://calendar.carleton.ca/undergrad/.

Students who have not satisfied the prerequisites for this course must either a) withdraw from the course, or b) fill in a prerequisite waiver (www.sce.carleton.ca/ughelp), or c) may be deregistered from the course after the last day to register for courses.
Carleton University  
Department of Systems and Computer Engineering  
SYSC 4101  
Software Validation, Verification and Testing  
Fall 2017

Course Outline

Lectures:
When: Tuesdays and Thursdays from 10h05 to 11h25.
Where: UC282.

Laboratory Sessions (compulsory):
When: Tuesdays from 14h35 to 17h25
Every other week, on even weeks, therefore Tuesdays on the weeks of Sept. 18th, Oct. 2nd, Oct 16th, Nov. 6th (the Fall break does not count), Nov. 20th, Dec. 4th
Where: CB5109
Students with conflicts: see the Undergraduate Calendar for regulations.

Contacts:
Instructor: Yvan Labiche  
yvan.labiche@carleton.ca
Teaching Assistants: Hoda Khalil  
HodaKhalil@cmail.carleton.ca
Wafa Hasanain  
WafaHasanain@cmail.carleton.ca

Office hours:
Yvan Labiche  
by appointment only  
ME4462
Hoda Khalil  
Wednesdays, 10h30—11h30  
MC7080  
(subject to change)
Wafa Hasanain  
Tuesdays, 12h—13h  
MC7080  
(subject to change)

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. Once 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).
Grading Scheme:
- four assignments, each worth 2% of the final mark;
- two (closed-book) midterm exams, each worth 25%;
- a (closed-book) final exam worth 30%.
- compulsory lab work worth 12%: each of the six labs is worth 2% of the final mark.

**To pass the course,** an appropriate overall mark (D- or higher) must be obtained, a passing mark (D- or higher) must be obtained at the final exam, and a passing mark (D- or higher) must be obtained for at least one of the two closed-book mid-term exams.

Important dates (subject to change):
- Assignment 1 will be posted on Sept. 8 and due on Sept. 29.
- Assignment 2 will be posted on Sept. 29 and due on Oct. 20.
- Assignment 3 will be posted on Oct. 20 and due on Nov. 17.
- Assignment 4 will be posted on Nov. 17 and due on Dec. 8.

- 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.

Assignments and Laboratories:
Assignments and laboratories will lead you through the use of software testing techniques, and are a good starting point when preparing exams. Portions of the work from each assignment may be reused and refined in subsequent assignments and/or laboratories. You are encouraged not to “write-off” any particular assignment or laboratory just because of its relatively low weight in the overall grading scheme.

You are encouraged to discuss issues when working on assignments or laboratories. However, you are expected to submit your own work for grading (unless otherwise specified). 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 before midnight (23h55) of the due date.
  - Late assignments will be graded according to the following policy: a 20% penalty per day (i.e., 24 hours) with a maximum of two late days (48 hours) after which the grade of 0 is assigned.
- Laboratory work is due at the end of the laboratory session.
  - Late laboratory work will receive a grade of 0.
- All submissions are on cuLearn. It is **your** responsibility to ensure that your material has been submitted. You must check that your material has been submitted. If no material is received, the grade of 0 is assigned (unless this was obviously a glitch in the online system, which will be investigated by the instructor and ITS).
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 6</td>
<td>Testing Object-oriented systems</td>
</tr>
<tr>
<td>Week 7</td>
<td>Testing a class, testing generalizations, …</td>
</tr>
<tr>
<td>Week 8</td>
<td>Regression testing:</td>
</tr>
<tr>
<td>Week 9</td>
<td>selection, prioritization, minimization.</td>
</tr>
<tr>
<td>Week 10</td>
<td>Revisiting stubs, drivers and oracles by examples.</td>
</tr>
<tr>
<td>Week 11</td>
<td>Testing real-time systems, distributed systems, testing GUIs</td>
</tr>
<tr>
<td>Week 12</td>
<td>Challenges and solutions.</td>
</tr>
<tr>
<td>Week 13</td>
<td>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: