CS4472A/9550A Course Outline Fall 2018

1. Course Information:
    - *Course Calendar*
        - Western Calendar Sessional Dates 2018; see also Western Multicultural Calendar
        - [W0] Thursday 6 Sept 2018: Undergraduate Classes Begin
        - [W1] Tuesday 11 Sept 2018: first class meeting
        - [W1] Friday 14 Sept 2018: Last day for late registration
        - [W2] Tuesday 18 Sept 2018:
        - [W3] Tuesday 25 Sept 2018: (warm-up exercise due end of Wednesday 26th of September 2018)
        - [W4] Tuesday 2 Oct 2018: (first testing exercise due end of Wednesday 3rd of October 2018)
        - [W5] Monday 8 Oct 2018: Thanksgiving Holiday at Western as well as undergraduate Reading Week. -- no classes this week
        - [W6] Tuesday 16 Oct 2018:
        - [W7] Tuesday 23 Oct 2018: (second testing exercise due end of Wednesday 24th of October 2018)
        - [W8] Tuesday 30 Oct 2018:
        - [W9] Tuesday 6 Nov 2018:
        - [W10] Monday 12 Nov 2018: last day to drop without academic penalty -- at least 3 days prior assessment for 15% of course mark should be returned.
        - [W10] Tuesday 13 Nov 2018: (third testing exercise due end of Wednesday 14th of November 2018)
        - [W11] Tuesday 20 Nov 2018:
        - [W12] Tuesday 27 Nov 2018:
        - [W13] Tuesday 4 Dec 2018: (fourth testing exercise due end of Wednesday 5th of December 2018)
        - [W13] Friday 7 Dec 2018: Classes End.
        - Time of Final Exam to be announced by Registrar's Office.
    - *Registrar's TimeTable (as of 30 Aug 2018):*
        - TC 203 Tu 7 - 10 pm
    - *Academic Calendar:*
        - Computer Science 4472A/B - Specification, Testing and Quality Assurance
        - Concepts and state of the art techniques in software specification and quality assessment for software engineering; quality attributes; formal specification and analysis; verification and validation.
        - Antirequisite(s): Software Engineering 4452A/B, Software Engineering 4453A/B.
        - Prerequisite(s): Computer Science 3307A/B/Y.
        - Extra Information: 3 lecture hours, 0.5 course.
    - *Senate regulation:* Unless you have either the requisites for this course or written special permission from your Dean to enroll in it, you may be removed from this course and it will be deleted from your record. This decision may not be appealed. You will receive no adjustment to your fees in the event that you are dropped from a course for failing to have the necessary prerequisites
    - *NOTE:* As a computer scientist, you should recognize the above as a case of patching flawed software (in this case the registrar's online enrollment software) with `blame the user' policies. Hopefully you will never do anything like that in your future undertakings.
    - *NOTE:* Also on the subject of flawed software, we won't be using OWL in this course. Instead, we will be using an Atlassian-based system which is more common in the CS workplace (and hence more useful to learn to use). It uses the git and ssh standards for file transfers as well as a blogging system that will contain our course annoucements and a mechanism for feedback and discussion on the pages of the system.

- *NOTE:* This course is cross-listed as a 95xx grad course. This means that graduate students make take this course for credit (see department specific rules on counting grad courses). It also means that the evaluation mechanism of this course is the same for graduate students and undergraduate students. And, of course, the meeting times and assignment times of the course follow the undergraduate sessional dates.
    - For a related 96xx grad course (Fall 2018, Monday afternoon) see: http://www.csd.uwo.ca/~webber/CS9622 -- Nonfunctional Software Requirements: Safety, Accessibility, and Sustainability.

2. *Instructor information:* Robert E. Webber, MC384, webber@csd.uwo.ca (include course number in subject line and send from your university email address), office hours on course announcements page and/or blog.

   Note email is not suitable for marks, which will instead be uploaded to your class git repository. Also note that the majority of my email is spam, and so reading email for me is mostly about deleting messages that don't come from strange addresses and/or don't have meaningful subject lines. Don't bother emailing attachments as for obvious security reasons, I don't open attachments -- if necessary, use your git repository to make available to me files that you might otherwise email. Also, use the course wiki if you have questions that are not specific to you, such as requests for clarification on assignments or exam questions -- I generally check the course wiki the first thing when I get in and am more likely to check it than my email during the day.

3. *Course Syllabus*
    - The goal of the course is to introduce students to modern software testing and quality assurance. In this course, skills learned will be evaluated via practice tasks and new vocabulary and concepts learned will be evaluated via the final exam.

      Ruby will be the main programming language for the course -- students are not required or expected to have prior Ruby experience. It turns out that the Ruby community has devoted a lot of effort to advocating and developing free tools to support various techniques in software quality assurance and testing -- making it ideal for practice tasks in this course. Most notable among the tools are the testing tool RSpec to promote Behavioral Driven Development; Mutant to evaluate the quality of testing; and Cucumber to promote Story Driven Development. Also notable are code quality tools like Reek (for detecting code smells) and Rubocop (for static code analysis). Both the RSpec and Cucumber systems can also be used for testing web applications via their interfaces to the Selenium web-application testing-framework.

      Course material will be made available on the course Confluence wiki and practice work handed in with Git to the course BitBucket Git repository (see course web page for relevant links). [Should the Atlassian software (Confluence and/or BitBucket) fail, we will back up to a course web page and paper reports as done in previous years]

    - The grading scheme is presented in the Methods of Evaluation section to both allow for course marks to be generated, but also to provide a framework that will allow students to continue practicing software testing and quality assurance and be able to self-evaluate.
    - The course format is to have assigned readings (or occasionally videos) that are available from the web (augmented by Western's digital library collection). Readings will be discussed in class and on the course blog. Class itself will focus on discussion of the reading material and how it might be applied in practice. This course is more interested in code reading and testing than in code writing. Time allocation for the course is based on 3 hours a week of class meetings; 3 hours a week of reading; and 3 hours a week of practice of course material.

4. *Course Materials*

- The course will make use of plain text files, pdf files, html files, and youtube video files that are freely available on the web or, at least, available to UWO students through the online subscriptions maintained by the University library system (and accessible from off campus via the Library proxy server as well as from on campus using machines with UWO IP addresses). For help with the Library proxy server, see help desk at any campus Library.
- *Course web page:* The course web page can be found at: http://www.csd.uwo.ca/Courses/CS4472a/ -- which is the same as http://www.csd.uwo.ca/~webber/CS4472 . On it are links to this course outline and a link to the course blog and practice area. If needed, it will become a backup of material normally kept on the blog such as: the schedule of material to be covered in the course and any announcements. Marking is done on the assumption that people check the announcements page of the course web page and blog frequently, and in particular, before handing stuff in.

5. *Methods of Evaluation*
   - *70% -- Practice Exercises*: Practice exercises are due in the git repository by the end of the day due. Note: it does not matter what time your computer says your copy of the file was created nor what time the git repository reports (which tends to use your computer's time), but rather whether or not the files were in the git repository when I downloaded the assignment files -- which I typically do when I come to school the day following the due date. There are no late extensions. If the due date is missed, for exercises counting 15% or more, it requires approval from the Dean's office to get an extension. For the warmup exercise, if you aren't going to be able to make the due date, contact me soonest.
     - 5% -- warmup exercise to make sure you are set up for the remaining tasks.
     - 15% -- first testing exercise
     - 15% -- second testing exercise
     - 15% -- third testing exercise
     - 20% -- fourth testing exercise

     The due dates for the exercises are lilsted in the Course Calender at the top of the course outline.

     Note: git allows you to hand in work multiple times (and the web interface to BitBucket lets you verify that you really did hand it in). You should use it to upload frequently. Then you get the benefits of both an off site backup (in case your computer dies or you accidently delete the wrong file) as well as being sure you didn't forget to hand in (something a few students seem to always manage to do). While there is a marking gap between having a `perfect' assignment and handing in a buggy one, the marking gap between a buggy assignment and no assignment is much larger.

   - *30% -- Final Exam*: The final exam is closed book (See Statement on Use of Electronic Devices). It will consist of a random sampling of short answer questions from a database of vocabulary and concept questions generated from the class material. The full database will be made available as much before the final exam as possible (I generally aim for a stable version a week before the final exam). Students are strongly encouraged to review the various versions of the database and indicate any questions that are awkward to read or whose answers they think are wrong. When the marking is done, it is the answers in the database that are viewed as the only correct answers. For many questions, the database contains multiple answers -- you are only required to know one of the various answers to a question, but if you write more than one answer to a question on the exam, then the question is wrong if any of the answers are wrong. There are no partial credits on these questions, they are either right or wrong.

     Generally people do well on this sort of exam, particularly if they take advantage of the opportunity to review the material before sitting the exam and getting any clarification needed on what the questions are saying. Although people often complete early, it is a mistake to not take the time to double check your work.

6. *Additional Statements*

- *Statement on Use of Electronic Devices:* The final exam (and any other quizzes or exams administered for this course) will be closed book, closed notes, with no electronic devices allowed, with particular reference to any electronic devices that are capable of communication and/or storing information.
- *Statement on Use of Personal Response Systems:* Clickers will not be used in this class as it is nearly impossible to use them and protect student privacy, as any Computer Scientist should know.
- *Statement on Academic Offenses:* Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site: http://www.uwo.ca/univsec/academic_policies/appeals/scholastic_discipline_undergrad.pdf
- *Statement on Academic Offenses (graduate version):* Scholastic offenses are taken seriously and students are directed to read the appropriate policy, specifically, the definition of what constitutes a Scholastic Offence, at the following Web site: http://www.uwo.ca/univsec/pdf/academic_policies/appeals/scholastic_discipline_grad.pdf
- *Regarding Plagiarism-Checking Software:* Such software will not be used in this class as it is nearly impossible to use them and protect student privacy, as any Computer Scientist should know. In the case of both material handed in and tests and/or exams, any plagiarism checking will be done the old fashioned way, by a person looking at the material in question.
- *Senate regulation:* Students who are in emotional/ mental distress should refer to Mental Health @ Western http://www.uwo.ca/uwocom/mentalhealth/
- *Senate regulation:* For UWO Policy on Accommodation for Medical Illness and a downloadable SMC see: http://www.uwo.ca/univsec/pdf/academic_policies/appeals/accommodation_medical.pdf [downloadable Student Medical Certificate (SMC): https://studentservices.uwo.ca/ under the Medical Documentation Heading] Students seeking academic accommodation on medical grounds for any missed tests, exams, participation components and/or assignments worth 10% or more of their final grade must apply to the Academic Counselling office of their home Faculty and provide documentation. Academic accommodation cannot be granted by the instructor or department.
- *Department specific information*
    - *CS Department Rules of Ethical Conduct*
    - *CS Department Specific Information on Scholastic Offences*

7. *Specification for this document*
    - *course outline requirements*
    - *requirements regarding undergrad evaluation*
    - *requirements regarding scheduling of tasks*