Notes re first class: Tuesday, 12 Sept 2017

Cover:

- http://www.csd.uwo.ca/~webber/CS4472/2017/CS4472CourseOutline2017.html
- Layout of course material:
    - Course Announcements
    - Practicing Testing Using the Ruby Programming Language
    - Quality Assurance for Better Course Performance
    - Test Driven Development -- The Relation of Software Design to Testing
    - How to Know If You Have Tested Enough -- Coverage and Mutation
    - How to Figure Out What the Result of a Test Should Be -- The Oracle Problem
    - How to Test a Property
    - How to Understand a Program
    - What Was the Program Supposed to Do, Anyway? Specifications
    - Preventing Program Errors rather than Finding Program Errors -- Quality Assurance
- The course uses the Atlassian Confluence and Bitbucket (git repository) systems to manage course material (Confluence) and practice work (BitBucket). This system was used in Spring 2017 by the CS3342 course. For more information, see:
    - Confluence 101: Getting Started With Confluence - Webinar 41:50 https://www.youtube.com/watch?v=y1YTsMTrC7c – their example is more complicated than the structure I use for the course.
    - https://confluence.atlassian.com/conf57/working-with-pages-701434708.html relevant Confluence documentation
    - https://www.youtube.com/watch?v=dCvjIo6nfHo a quick comparison between BitBucket and GitHub
    - People whose registration had made it to the registrars computers last week, should find they already have access by class time on Tuesday. People whose registration is still being processed should check again later this week. The registrar is regularly updating their information with our system staff who are regularly updating the course wiki and Bitbucket pages.
    - Bitbucket - the Git solution for professional teams https://www.youtube.com/watch?v=BD8xfCILcBs 1:26
    - note: The version of Bitbucket we will be using in class is running on a local server and is not the version that runs at Atlassian. Information on how to connect to your private course repository will be posted in the confluence wiki under https://wiki.csd.uwo.ca/display/CS44722017/Course+Announcements .
    - note: in order to use Bitbucket, you will need git to be installed on the machine you are working on. On many systems, git comes preinstalled. But if it isn't on your machine, then visit https://git-scm.com/ and, in particular, https://git-scm.com/downloads . Note: if you are running a linux shell under Windows 10, then you want git to be running in the shell rather than in the Windows space. For the most part, once you have set up your connection to the course Bitbucket area, your work will be done using git commands and you won't have to worry much about the difference between BitBucket and other repositories.
- Initially the course practice work will be focussed on using Ruby and various tools that work with Ruby.
    - If you haven't used Ruby before (or want a quick review), I recommend:
        - Ruby Programming -- Learn the Ruby Programming Language in One Video (talk) https://www.youtube.com/watch?v=Dji9ALCgfpM 55:29 code at: http://www.newthinktank.com/2015/02/ruby-programming-tutorial/
        - visit https://www.ruby-lang.org/en/ that not only has download links, but also under Getting Started, a number of short tutorials.
        - see also: https://www.ruby-lang.org/en/documentation/
        - once you get a sense of how things work, you can generally get by with http://ruby-doc.org/core-2.4.1/ and http://ruby-doc.org/stdlib-2.4.1/
        - but don't forget, Google and StackOverflow are your friends, as in queries like: https://www.google.ca/search?q=stackoverflow+How+do+I+concatenate+strings+in+Ruby
    - If you already know Ruby, the next thing of interest is the tools we will be using with Ruby. We will look at three testing frameworks, the first is minitest https://github.com/seattlerb/minitest .
    - There is also the question of whether or not you are writing reasonable code in Ruby. While no program can solve this problem completely, Ruby has a very good tool for checking code quality called
        - reek: a code smell detector: https://github.com/troessner/reek
        - flay: looks for duplicate code: https://github.com/seattlerb/flay
        - flog: code metrics showing worst written code: https://github.com/seattlerb/flog
        - rubocop: enforces Ruby Style Guide: https://github.com/bbatsov/rubocop
        - rubycritic: https://github.com/whitesmith/rubycritic (I haven't tried this one out, but it seems to be a front end to reek, flog, flay, and rubocop)
    - In general, we will be looking at setting up a coding approach known as Test Driven Development (TDD). To see where we are headed, you might find the following of interest https://semaphoreci.com/community/tutorials/getting-started-with-minitest .
- **Note:** there are a lot of videos and tutorials and such that one can spend time reading. Reading such material is good, but it doesn't count as practice work any more than reading about piano playing or watching a piano performance counts as practicing the piano. In the next week or two, I will start posting practice exercise that you will be attempting during your practice time. The important thing is to practice and get better, it is not a raise to see who can play a scale the fastest.
- **Note:** the initial exercises will be non-web tasks. but in the coming weeks, task involving the web will be available for people who interested in that (as Ruby has much support for web applications). by mid semester, there should also be tasks to guide people in transferring what they have learned about testing with Ruby tools into languages they might be more interested in – however, for this to happen, people need to let me know in a month or so what situations and environments they would like to see testing tasks on.

Items to look at for next class:

- Hmm. Much of the material above starting with the mention of Atlassian is stuff we will talk more about next class and so count in this category.
- For people interested in some talks we will be looking at later in the semester on testing, I recommend:
    - RailsConf 2014 - Mutation Testing with Mutant by Erik Michaels-Ober https://www.youtube.com/watch?v=WccaOMuf01Y
    - John Hughes - QuickCheck Evolution https://www.youtube.com/watch?v=gPFSZ8oKjco
    - Exploratory Testing https://www.youtube.com/watch?v=8Y4WCdJRfV4