Study Guide for Tests

Quiz 1 - Friday 2/22/19

Test one will cover all of the lectures and labs we have done so far up to and including Lab 5 (See Spring2019Schedule.md for more info), but will not cover Scientific Computing. The format will be some short answer questions and some free response questions. You can refer to (see previous example test posted in the slack channel). Review all lecture/guest lecture notes and all labs. Make sure you have a solid understanding of how Github works.

For this test I would recommend having the following software installed beforehand.

- Python
- Latex
- git
- C/C++ compiler

If you have gone through the labs up to this point, you should have most of this already.

Remember that this test is open notes (but closed internet). Make sure you have a recent version of the repository cloned: https://github.com/rcos/CSCI-49XX-OpenSource.git and that you have local copies of any reading/reference material you may want to use. You may also bring your notes.

Make sure all of your lab code is on your local computer as you are allowed to refer to it during the quiz and may find it helpful to do so.

Any topic we covered or that was assigned as reading is fair game, but specific topics that I would review for the quiz are:

- Who are the major personalities behind open source/free software? How did the movement evolve? What are its connections to copyright and other property laws?
- What are the basics of Open Source (What is the definition of Open Source? What are the Four Freedoms of Open Source? What 10 characteristics comprise open source software?)
- What are the characteristics of a permissive open source license (BSD, MIT), a copyleft open source license (GPL, LGPL).
- Know the major characteristics of MIT, BSD, AGPL, GPL, LGPL, Mozilla and Aache licenses and know how they impact your ability to create, use, and distribute software.
- Know make and CMake and what is going on when you use them (Review Build Systems Lecture)
- Why is it important to have community?
- Why is documentation important?
- Why is testing important?

- Given a program, implement a new feature or add unit tests (know how to do both)
- Know how to analyze a program (describe what is going on in a program)
- Know the basic flow of github/git from the command line (Review GitHub Tutorial)
  - Basic flow is add->commit->push
  - Know how branching works and what checkout command does
  - Know how pull requests works
  - Know what is a merge conflict, how it can arise and how to resolve it.
  - Be able to set up remotes and create workflows
- Know how to solve and write Regular Expressions (RegEx) (Review RegEx Tutorial)
- Given a mathematical expression or equation know how to write it in Latex

If you have any more questions feel free to contact the Undergraduate Programming Mentors or Professor Turner (preferably on the Mattermost channel) before the test! Good luck everyone!

Sample Questions -

Find the bugs in the testing program. If you want to run this, compile using with the -std=c++11 flag (g++ -std=c++11 testing.cpp -o main.exe). How could you write unit tests to easily identify these bugs? Write a prev and next function and write unit tests for them. Prev/Next should choose the previous and next days from the starting vector based on the current day selected (Be careful about edge cases!). For example if I have "WEDNESDAY" currently selected prev() should give me "TUESDAY" and next() should give me "THURSDAY". If I have "SUNDAY" selected next() should give me "MONDAY" and if I have "MONDAY" selected prev() should give me "SUNDAY". Write unit tests for prev/next/randomize functions (think about what the program is changing and how to check it). Specifically write tests to show:

- 1 day from a SUNDAY will be a MONDAY
- 2 days before a TUESDAY will be a SUNDAY
- 40 days from a FRIDAY will be a WEDNESDAY
- 87 day before a WEDNESDAY will be a SUNDAY

Write the following formulas in Latex (optional integrate the integral ignoring the y = 0 and y = 5 limits and write the formula for it using c for the constant)

Formulas

Write a regex expression to check if a string contains only AlphaNumerical (A-Z, a-z, 0-9) characters

How would you resolve the following merge conflict (pick whichever one you feel is correct)-

<<<<<<< HEAD To whom'st'd've'ed it may concern, ======= To whom it

may concern, >>>>>>> da18a796d6f3715b9bb0c58b8ef63a96aebf8511 Hello from the other side I must have called a thousand times

What git commands would you run after resolving this conflict (Let's assume this file is called Adele.md)?

Solutions are here - Look at comments at the bottom of the file.

Quiz 2 - Tuesday 4/16/19

Test two will focus on material from *Open Source in Scientific Computation* up through TensorFlow (Labs 6 through 11), although, you are expected to be able to apply the material from earlier in the semester as well. (See Spring2019Schedule.md for more info). The format will be some short answer questions and some free response questions. I do not yet have a sample test, so you will need to refer to Test 1 for the testing style. Review all lecture/guest lecture notes and all labs, particularly Labs 6-11.

For this test I would recommend having the following software available.

- Networkx
- matplotlib
- R
- ctest
- mongo
- TensorFlow, numpy, python-tk, pillow and keras
- Docker

If you have gone through the labs up to this point, you should have most of this already.

Remember that this test is open notes (but closed internet). Make sure you have a recent version of the repository cloned: https://github.com/rcos/CSCI-49XX-OpenSource.git and that you have local copies of any reading/reference material you may want to use including for the listed packages. You may also bring your notes.

Make sure all of your lab code is on your local computer as you are allowed to refer to it and may find it helpful.

Any topic we covered or that was assigned as reading is fair game, but specific topics that I would review for the quiz are:

- Why is testing important, what is a dashboard, and how do you add tests to an existing project using CMake?
- How do you execute CRUD (Create, Read, Update, and Delete) operations in Mongo? Make sure you revisit this from Lab. Some of you did not update records correctly and were unable to pull them back with a subsequent query.

- What tools are available for doing scientific and statistical programming using open source tools? Be able to use networkx, python and R to answer some simple questions.
- Be able to run and interact with a Docker container. Be able to use a Dockerfile and/or a YAML file to create a container with specific requirements.
- Be able to create test data, test, and verify a TensorFlow. Know the varoius terms we used: Test, Training, loss, epoch, Neural Net, etc. and be able to describe them.

If you have any more questions feel free to contact the Undergraduate Programming Mentors or Professor Turner (preferably on the Mattermost channel) before the test! Good luck everyone!