CS 5154
Course Project Kick-off

Owolabi Legunsen

Spring 2021

Course Project

- Work in small teams to test open-source software

- Goals:
  - Gain deeper knowledge and expertise than we can cover in class and homework
  - Learn how to test (parts of) existing and large software
  - Learn cutting-edge tools techniques and tools and apply them to open-source code

Three Course Project Tracks

1. Test commons-math and openmrs

2. Test some open-source software that you care about
   a. You will be responsible for finding a team
   b. You may get less guidance

3. Develop, improve, or test open-source testing and analysis tools
   a. I will provide some suggestions
   b. You will be responsible for finding a team
   c. May be very challenging

Open-source code you may test

- Apache commons-math
  - https://github.com/apache/commons-math

Apache Commons Math

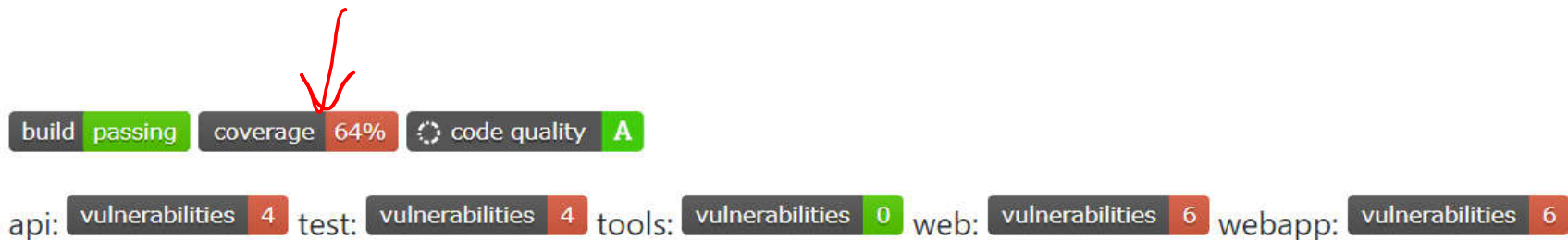`build passing` `coverage 91%` `maven central 3.6.1` `license apache`

The Apache Commons Math project is a library of lightweight, self-contained mathematics and statistics components addressing the most common practical problems not immediately available in the Java programming language or commons-lang.

Some code that you may test (2)

- OpenMRS core
  - https://github.com/openmrs/openmrs-core

build passing | coverage 64% | code quality A

api: vulnerabilities 4 test: vulnerabilities 4 tools: vulnerabilities 0 web: vulnerabilities 6 webapp: vulnerabilities 6

OpenMRS is a patient-based medical record system focusing on giving providers a free customizable electronic medical record system (EMR).

The mission of OpenMRS is to improve health care delivery in resource-constrained environments by coordinating a global community that creates a robust, scalable, user-driven, open source medical record system platform.

Project Sprint 0.1

- Assigned on CMS
  - Due on 3/8/2021 at 9:30am EST

- Goals:
  - Get more familiar with Maven, OpenMRS, commons-math

- Work in randomly-assigned teams of 3-4
  - Perhaps you can find project teammates from this sprint?

Tentative timeline for course project

- Detailed plans: TBD

- Project kick-off date: ~~2/24/2021~~ **3/1/2021**

- Final project report due: 5/14/2021
  - At least three other reports will be required during the semester

Likely project timeline

Tentative

Phase	Objectives	Start	End	Report
Sprint 0	Play with open-source projects	3/1	3/8	no
Proposal	Find track, initial team, project	3/8	3/15	yes
Phase 1	Apply test-driven design	3/15	4/5	yes
Phase 2	Use mutation testing, test generation	4/5	4/28	yes
Phase 3	Use regression testing, runtime verification	4/28	5/14	yes

- Only Tracks 1 and 2
- Tentative

The project will overlap with HWs

Tentative

No	Topic	Start	End	Team
0	Warm-up	2/8	2/15	no
1	Input Space Partitioning	3/8	3/15	yes
2	Graph-based Testing	3/15	3/22	yes
3	Logic-based Testing	3/22	3/28	yes
4	Syntax-based Testing	4/5	4/12	yes
5	Test Generation and CFG construction	4/26	5/3	yes

- There will be pockets of time with no homework
- You should regularly be working on your course project

How will teams be formed

- Projects (3-4 persons per team)
  - You can suggest who you want to be in a team with (and why)
  - No guarantee that you'll get who you want
  - Pseudo-randomly assign folks who have no suggestions
  - I reserve may modify/rotate the teams at intervals

- HW (2-3 persons per team)
  - Will be mostly randomly assigned
  - I may rotate teams every other HW
  - IRL: you almost never choose your team

More participation *point* opportunities

If you notice some bugs related to the course (on web page, slides, code, tests, text…) any time during the course, please report them to Owolabi, and you can get participation points!

Questions?