CS 193S: Assignment #1

Due Thursday January 21, 2:15PM

This is an individual assignment. Take the GWT sample application and convert the initial user interaction into a unit test (perhaps more correctly an integration test) using the JUnit framework (JUnit 3 is what works for GWT). Then, write unit tests and code to display a randomly generated 20 digit number in one or more original and interesting ways, using an external API, such as a Maps or Search API, as well as an existing open source GWT based or java/javascript library.
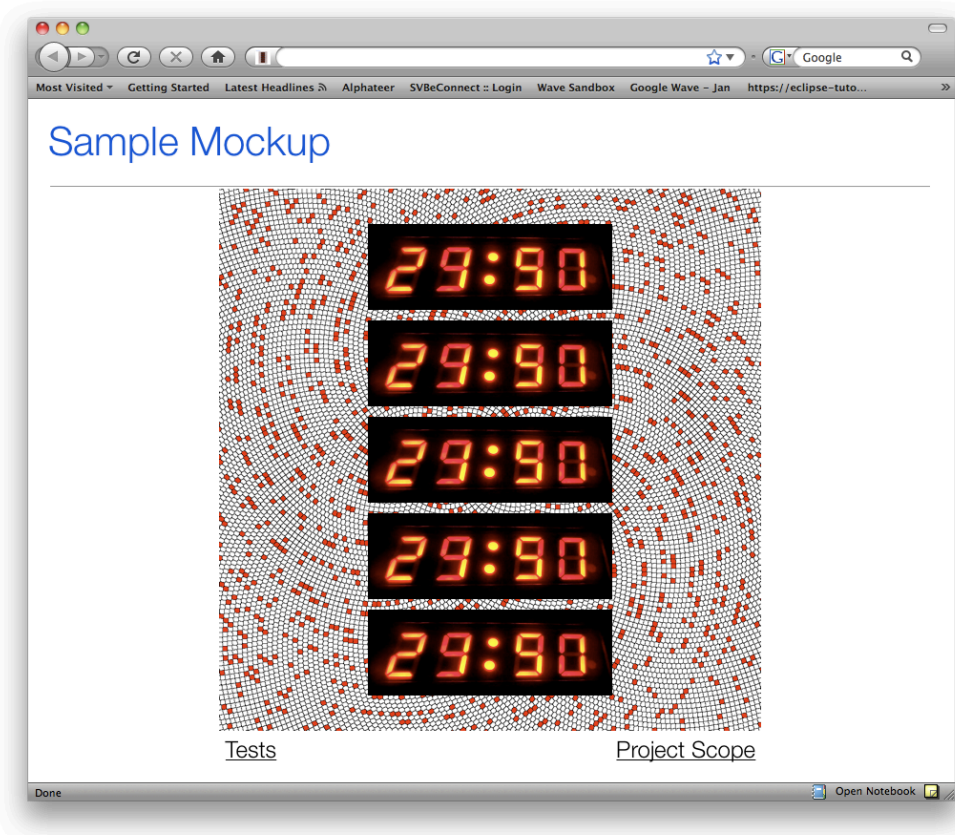
For example display the number "98765432109876543210":

• as a captcha,
• as a product of prime factors,
• within a web counter,
• etc.

The tests can be run by clicking on a link that is displayed on the main page. The test page should display the names of each test and the resulting output. So you should deliver a web page that displays a 20 number in a very creative way, and a link to a test result display page, which when clicked, runs the tests and displays the output of the tests.

Finally, add a display that briefly (1–2 paragraphs) describes the type of website project you

would like to work on this quarter. You should imagine these paragraphs to be a way that other students can decide if they'd like to work together with you on a project.

For extra credit, publish the code to appengine, so that other students can see it run.

CS 193S: Assignment #1

The grading of the assignment will focus on correctness of the basic code and the tests and the coverage of the tests. Are corner cases handled? Are the public methods tested? Does the code connect to an external API? Does it make use of an additional library?

The originality of the display and the code it connects to will be used to help define teams for the larger programming project, and which projects will be candidates for demonstration to an angel investor group. In the process of completing this assignment, feel free to share ideas, strategy, advice with your classmates and colleagues, but don't share code.

Project teams will have one or two advocates who help provide team cohesiveness, a few contributors who can work on team projects, and a few 'consultants' working on individual projects. The structure of the teams is such that people can work on tasks individually as consultants, or as more closely knit coding groups.

Some useful documentation can be found here:

http://java.ociweb.com/mark/programming/GWT.html