Individual Assignment #2

Due Date: Friday, August 11, before 10:00pm

Tasks:
This is an *individual* assignment – all questions should be answered by you, according to your thinking.
1.) If needed, familiarize yourself with the purpose of doing reflective essays (attached).
2.) Answer the following 7 questions:

Q1. *This question concerns the topic of unit testing. (You may find the several articles under "Useful Resources" on the course web useful as a way to approach the problems below.)*

Q1.1. *Write a simple Java class for representing a rectangle. It needs to have public methods computing its perimeter and area. Now write unit tests for that class.*

Q1.2. *Imagine being hired as a consultant in a company to develop unit tests for their legacy code, some of which, unfortunately, comes without a specification. Consider the following Java class left behind by a programmer who is no longer with the company.*

```
import java.awt.geom.Point2D;

public class TOS {
    private Point2D.Double p1, p2, p3, p4;

    public TOS(Point2D.Double p1, Point2D.Double p2,
               Point2D.Double p3, Point2D.Double p4) {
        double lp1 = Math.sqrt(p1.x * p1.x + p1.y * p1.y);
        double lp2 = Math.sqrt(p2.x * p2.x + p2.y * p2.y);
        double lp3 = Math.sqrt(p3.x * p3.x + p3.y * p3.y);
        double lp4 = Math.sqrt(p4.x * p4.x + p4.y * p4.y);
        this.p1 = new Point2D.Double(p1.x / lp1, p1.y / lp1);
        this.p2 = new Point2D.Double(p2.x / lp2, p2.y / lp2);
        this.p3 = new Point2D.Double(p3.x / lp3, p3.y / lp3);
        this.p4 = new Point2D.Double(p4.x / lp4, p4.y / lp4);
    }

    public double tos_Obik() {
        return tos_DOC(p1,p2) + tos_DOC(p2,p3) + tos_DOC(p3,p4) + tos_DOC(p3,p4);
    }

    private double tos_DOC(Point2D.Double a, Point2D.Double b) {
        return Math.acos(a.x * b.x + a.y * b.y);
    }
}
```

Write unit tests for it. How do you know that what you're testing was the intended behavior?

Q1.3. *In addition to your code and the answer to the question in* **Q1.2**, *submit a screen snapshot showing the process of running your unit tests in a development environment of your choice.*

Q2. *Give an example / snippet of your own code from the current project that needed (or still needs) refactoring. What is the motivation for refactoring there? Now show how you have refactored the code. In what ways did your refactoring improve the code?*
<u>Note</u>*: Refactoring will be the topic of an upcoming lecture early next week.*

Q3. *Show your team's latest version of the application you're building to two outside people and let them "play," while you observe and take notes. Discuss their main comments and what usability issues they uncovered. Did you learn additional information from the second participant? (Your description should be no more than 1 page.)*
<u>Tip</u>*: For some suggestions on usability studies, refer to "Guidelines for User Testing with Thinking Aloud" from the "Useful Resources" section of the course web. Note that these are guidelines for a more complete*

usability study. For this assignment, we don't ask you to have a pilot study or a background questionnaire. We recommend that you encourage your users to keep talking while "playing." To do that effectively, it is best to use unobtrusive comments that don't point them in a particular direction.
Note: You do not need to recruit two observers for every team member – it is okay to meet as a group with, say, 5 users and then each team member can choose to discuss his/her observations on the reactions of two of these individuals.

Q4. *Consult the handout on Influence diagrams and look at the relevant slides with ink from lecture 15. Sketch an influence diagram (different from those described in the lecture and in the handout) that has at least 3 states and contains a positive feedback loop leading to an undesirable outcome. We suggest that you use this opportunity to illuminate an issue of interest to you personally; it does not need to be related to software.*
Briefly describe the issue in words too and discuss how one might "untie" the feedback loop specifically for the situation at hand (rather than in general).

Q5. *Read Joel Spolsky's short article "The Joel Test: 12 Steps to Better Code," linked from the course web under "Useful Resources." Which of the suggested twelve "steps" does your team faithfully practice on a daily basis? For the remaining practices, what hurdles have prevented the adoption of in your project?*

Q6. *Thinking back over your experiences in the team and on the project so far, list two aspects that have worked particularly well and two that have not worked so well for you. Briefly relate specific examples. For the aspects that were not so successful, what would you do differently next time you encounter a similar situation?*

Q7. *The usability design lecture revolved around principles to follow for creating usable objects. Poor designs often neglect one or more of these principles.*
Describe an example of software that you personally find difficult to use. Analyze the difficulty of use – can you relate the difficulty to any of the usability design principles? How would you make that software easier to use?
Tip: We suggest that you choose a very specific example, such as a dialog box or a feature that you have difficulty with, as opposed to a full application, such as Microsoft Word. Showing a snapshot of a particular interaction you're describing may help to additionally clarify your point.

Format:

- Use at most 4 pages of text *total* for all of your answers. Non-textual artifacts (e.g., images and code), if any, are limited to 2 additional pages (embedded inside the same document).
- Format your document to be single-spaced, using font size 11 or larger.
- Save your work preferably in rich text format (RTF or DOC). We cannot edit (and add comments to) PDF without copying the contents over.

Things to remember:

- Name the file that contains your essay using the following convention:
  - *LastName*-cse403-assn2.rtf, where *LastName* is replaced with your last name.
- Put your name also in the header field of your document, not only in the filename.

Submission: via UW Catalyst's eSubmit tool:
https://catalyst.washington.edu/webtools/secure/esubmit/turnin.cgi?owner=vrazmov&id=4670

Follow-up: As part of our feedback, we will pose a follow-up question or two, based on what you wrote. You will be expected to answer that in writing too, though we anticipate that the answer will take you less time to produce than the original writing did.

Reflective Essays

The goal of reflective essays is to:

(a) get you to think more deeply about certain aspects of the course;
(b) give you an opportunity to practice articulating your reflections.

Reflecting upon your experience deepens your understanding of the particular domain, shows it in a different light, and helps you to generate new ideas and possibilities in this and other domains. Practicing reflection is important, since this skill is critical for becoming an expert in *any* field – only by reflecting upon what happened can you learn from previous successes and mistakes.

Here are some general techniques that will help you to get the most out of your reflections:

- Write in the first person about what happened to *you*, not to some abstract person.
- Write about things that are meaningful to *you* or happened to *you*. You will have more energy to dig into those topics.

Consider the following aspects as you reflect upon your experience:

- What stood out for you?
- What insights did it give you?
- What is *one* thing you will do differently as a result of what you learned?
- Use facts to ground your assertions.
- Do not confuse opinions with facts.

We are looking for stories about issues meaningful to you, communicated in a way that makes them meaningful to the reader (i.e., us).

Good essays have solid content, clear style, are grounded in factual assertions, and communicate well. Incidentally, good code has the same characteristics, so excellent software developers tend to also be excellent writers. In the end, writing code and writing a good reflective piece are both about communicating effectively with your audiences – an indispensable skill regardless of your domain of expertise.

In our experience, good reflective essays often include one or more of the following aspects:

- Demonstrate how lessons from this course have enabled you to do things that you were not able to do before – things not necessarily related to the course. This may include noticing aspects of the world around you that you had not previously noticed.
- Discuss how you have used lessons from other areas of your life to benefit your (or your team's) performance in this course.
- Illustrate how this course is (or is not) changing your beliefs and ideas about what is (or is not) possible for you to achieve.
- Analyze situations using tools / techniques learned in this course, and discuss how well those tools / techniques worked for you.
- Discuss a situation from multiple perspectives.