

# TCSS 360 Syllabus

## 1. THE BASICS

Instructor: Josh Tenenberg

Office Hours: by appointment

Instructor email: jtenenbg@uw.edu

Course homepage: Google "Josh Tenenberg" and follow the links from my homepage.

## 2. PREREQUISITES

The prerequisites for this course are a minimum grade of 2.0 in TCSS 342 and 10 credits of writing coursework. Please see me immediately if you do not satisfy these prerequisites.

## 3. TOPIC OVERVIEW

Much software is written poorly: late, over budget, buggy, and difficult to maintain. In this course, you will study how to build quality software using sane development practices while developing a broad conceptual understanding of software development.

The course topics include:

1. Software lifecycle and process models such as waterfall and agile methods
2. Effective groupwork practices, such as mutual accountability and public commitments.
3. Software development tools such as version control and unit testing frameworks.
4. Standard representations for requirements, such as user stories and business rules.
5. Design generation, design representation, and heuristics for good design.
6. Dynamic software verification: unit, integration, regression, and acceptance testing.
7. Static software verification: reviews, inspections.

## 4. STUDENT LEARNING GOALS

The student learning goals for the course are to:

1. Write and validate client requirements incorporating standard forms of representation for small projects of approximately 3 users carrying out several tasks each.
2. Construct and verify a design based on client requirements composed of multiple computational units, represented with standard forms.
3. Code and test a subset of client requirements consistent with a design.
4. Participate effectively in a team to carry out the above tasks using appropriate tools for facilitating teamwork.

The CSS Degree Student Learning Outcomes that this course contributes to are:

- a. an ability to apply knowledge of computing and mathematics appropriate to the discipline;
- b. an ability to analyze a problem, identify and define the computing requirements appropriate to its solution;
- c. an ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
- d. an ability to function effectively on teams to accomplish a common goal;
- e. an understanding of professional, ethical and social responsibilities;
- f. an ability to communicate effectively with a range of audiences;
- g. an ability to analyze the impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues;

- h. recognition of the need for, and an ability to engage in, continuing professional development;
- i. an ability to use current techniques, skills, and tools necessary for computing practice.

The UWT Student Learning Goals that this course contributes to:

#### *Inquiry and Critical Thinking*

Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.

#### *Civic Engagement*

Students will define their roles and responsibilities as members of a broader community and develop an understanding of how they can contribute to that community for the greater good.

#### *Communication/Self-Expression*

Students will gain experience with oral, written, symbolic and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration with others to solve problems or advance knowledge.

## **5. COURSEWORK**

Each assignment is to be handed in no later than the start of class on the specified due date. Late assignments will not be accepted. With due cause, exceptions to this policy will need to be negotiated with the instructor; emergency documentation may be required and/or late penalties may be given, depending upon the circumstances.

The assignments and grade weightings are as follows.

- Project deliverables: 66%, with each deliverable weighted equally.
- Project check-ins: 10%, with each check-in weighted equally.
- Weekly Reports: 10%, with each report weighted equally.
- Reading Analysis on Object Oriented Design: 7%
- User story homework: 7%.

Note that there are also participation requirements, detailed in the page associated with groupwork. I reserve the right to make small adjustments to grade weights, or to add/remove assignments as the need arises. The specifics of each assignment are detailed in separate web pages linked from the course homepage.

## **6. GRADING**

Unless otherwise specified, each assignment will receive an integer score between 0 and 4, inclusive. Some of the assignments are graded high pass, low-pass, no-pass (4, 2, and 0, respectively). Your grade on each assignment will be a weighted sum of the grade on each part. Your final grade for the course will be calculated by taking the weighted sum of grades on all work that you have handed in, and rounding to the closest 1/10th. That is, multiply each score that you receive by the weight of the assignment, add these together, then round to the nearest 0.1. I will provide you with a grading template that you can use to track your progress throughout the term – just plug in the grades that you receive on each assignment.

Many students are accustomed to other instructors who grade out of 100, where an A is above 90, a B is above 80, a C is above 70, etc. These students sometimes believe that a grade of “2” that I assign (or ½ the numerical score of the highest grade of 4) is equivalent to getting a 50/100, and is therefore not a passing grade. This is incorrect, though. I use the UW policy for corresponding letter grades to numeric grades: 4

is an A, 3 is a B, 2 is a C, 1 is a D, and 0 is an E/F. For additional details, see UW Student Governance and Policies, Chapter 110, <http://www.washington.edu/admin/rules/policies/SGP/ScholRegCH110.html>

## **7. USE OF ELECTRONIC DEVICES**

Laptops may be used in class only to carry out course-related work at designated times. Please take phone calls (texting, checking messages, ...) outside the class.

## **8. PLAGIARISM**

This class will heavily involve the use of the written works of others. Your own written work will involve discussing the ideas of others. When using the ideas of others, it is important to acknowledge whose ideas you are using, and to clearly distinguish the ideas of others from your own. To convey the impression, whether inadvertently or deliberately, that another's work is your own, is called plagiarism. Plagiarism is a serious offense in the university. I have written a guideline on plagiarism and how to avoid it (<http://faculty.washington.edu/jtenenbg/courses/general/plagiarism.html>), and I expect that you will abide by it. Although this guideline is geared toward the use of other's computer programs, it applies equally well to other kinds of intellectual objects.

## **9. CLASS EMAIL LIST**

I have established a course mailing list for course-related discussion, to which everyone in the class will be a member. The email list name will be broadcast at the start of the course. Emailing to this address will send an email to every person in class and to the instructor.

## **10. STUDENT SERVICES**

Campuswide support for students is provided by Student and Enrollment Services. These include student health services, the counseling center, health and wellness education, and disability support services. See the following for the Student and Enrollment Services website:

<http://www.tacoma.washington.edu/studentaffairs/SHW/>