Welcome!
I and the entire teaching staff for CSE8B are incredibly privileged to welcome you to this class. In CSE8B we look forward to helping you take your next steps in developing proficiency in applying computational concepts to solve problems of interest and in developing skills to enable you to become life-long learners of programming languages.

What does this course offer to me?
This course helps you gain expertise in professional software development practices. There are two specific areas of concentration: A) computational techniques and programming language features and B) self-assessment skills and solo programming.

By now, you have discovered the power of basic computational features to direct the computer to do cool things, like mirror an image or attain a “greenscreen effect”. You have also experienced the basics of object-oriented design by organizing in a “fields with methods” approach. In this course, we’ll begin to ask you to explore additional computational techniques and design issues and to be able to compare and contrast them to the set you already know. You will gain professional level software development abilities as you learn to apply the best design and computational approach, rather than simply the one you are most familiar with.

Adopting professional standards and the ability to discuss and critique design of a problem solution is critical in today’s computational world. As the power of computation is more cheaply and ubiquitously available, the potential to solve more complex problems of interest grows. Quickly, we can find ourselves drowning in complex, hard to read code. Applying professional standards in use of program language features and design is key to developing, maintaining, and communicating about code in the team-based software development environment that is industry standard.

How, exactly, am I going to accomplish this?
This class has the following components which seek to support your learning:

- Constructive, activity-based lecture using clickers. Before each class period you will utilize a number of resources (reading of the two textbooks, reading of web-based materials, engagement with on-line activities or tutorials, viewing of videos, etc.) to develop basic understanding with the concepts of the lecture. Then, in lecture, the instructor will present questions and scenarios with which you will engage to test and further refine your understanding of the day’s concepts. You will test out your understanding through application of the “solo vote, discuss, group vote” peer instruction process. Difficult concepts (or those with lesser detailed textbook preparation) will be augmented by mini-lectures.

- Weekly assessment opportunities to gauge your individual understanding. Each week you will either complete an in-class quiz or an “on-line” quiz activity (PeerWise) designed to help you determine “what you need to know” and “how well you in fact know it” for the week’s material.

- Every two week solo programming assignments. You will be given the opportunity to fully test your personal understanding of concepts by developing more complex (and LONGER)
programming solutions to interesting problems. These programming assignments are to be completed individually. Compared to CSE8A, you are expected to program by yourself and to rely less on tutors or others to design, implement, debug and test your programs. Specifically, you may ONLY ask other students in the class questions about specific errors or bugs. You should seek to keep your code confidential from other students as much as possible. If you have questions beyond a well-confined bug, you should ask a tutor.

- **Discussion section.** Provides much of the opportunity to get further clarification about programming assignments and to get additional review or clarification on concepts that continue to be confusing after lecture.
- **Web-based support.** Using UCSD’s WebBoard system, you should develop proficiency in communicating electronically at a professional level regarding program specifications and programming problems. This is reflective of standard professional practice – both within an industry team and when seeking information from the broader world community.

Note that there is no specific “lab” (like CSE8AL) to support you in obtaining “first try” experience with the new computational features you will learn in this class. “Playing around” with concepts and language features to explore their basic behavior is something that we now consider part of your personal study process. We highly recommend that you do this A) in conjunction with textbook reading and B) with a study partner.

The specific content areas you will be addressing in developing more professional programming practices include:

- Computational problem solving approaches and techniques: recursion, file input/output and text parsing
- Expanding and refining your mental model of Java: reference behavior, parameter passing rules.
- Professional language features: wrapper classes, static methods, main tester methods, exception handling, inheritance, polymorphism
- Data representation issues: dynamically sized arrays

**How will I (and the professor) know if I am making progress in my learning?**

Responses on individual in-class clicker questions and performance on in-class quizzes will be your primary indicators of adequate progress in your learning. Note: *Compared to the weekly assessment data available to you last term in CSE8A, you will be MORE responsible for self-assessment of your progress*. If you experience any concerns, you should seek help immediately (do not wait a week). To help support you in this, we are providing access to the PeerWise system – an online quiz question submission system that you will use in weeks not having an in-class quiz. You may use PeerWise to get additional practice and to encourage you to consider (and develop) “good quiz questions” which reflect the week’s learning. (More details in handout).

The other key indicator of your progress in learning will be your individual programming assignments. Because programming assignments are due every 2 weeks, you will need to start on them EARLY (each can be expected to take you 10-12 hours). You will need to self-assess your capabilities as you work on these assignments (rather than waiting the 3 weeks to complete it and get a grade back). You should be asking yourself if you understand the specification of the assignment, if you can apply the basic concept required, if you have adequate debugging ability to sensibly (rather than randomly) debug your program, and whether you can explain what the different parts of your program do.