Welcome to CS253. I am Prof. Bruce Draper, and I will be teaching this class. You should also get to know your two Graduate Teaching Assistants (GTAs), Jason Yu and David White.

Let me start with some quick, procedural notes. Lectures are only 150 minutes a week. That means we can’t afford to waste 5 or 10 minutes of every lecture getting started. Lectures will begin at 9:30 sharp. That means you should be in your seat ready to go by 9:30, not shuffling through the door looking for a place to sit. There is an old adage: “to be on time is to be late; to be early is to be on time”. In return, I will end class by 10:45, so that you can get to your next class, appointment, or what have you.

As you may have noticed, this is a large class, with over 100 students registered for it. To make things worse, I am truly bad at names. I can barely remember my own kid’s names sometimes. So how am I going to learn all your names? I won’t, but I will learn as many as I can. To help me, I will ask you your name whenever you speak in class, whether you are asking a question or I called on you.

The interesting part of today’s lecture is why? Why are you here? But you have a quiz due before the next class, so I have to go over the administrative aspects of the class first. Let me start with what, when and how much before we get to why...

This course has reading assignments (with quizzes), recitations, programming assignments and tests. Let’s go over these:

- **Reading assignments & Textbooks**
  - The Weiss text is mandatory. This is what many of the lectures are structured around.
  - The Stroustrup text is optional – but I use it as a reference. The language specification is available on the web, but Stroustrup provides more explanations. If you use the web instead, please use www.cplusplus.com (you can find the link on the resources page of the class web site).
  - Reading assignments are posted on the web (progress page). They are listed by the day they are due. For example, by Thursday you need to have read Chapter 0 of Weiss.

- **Quizzes**
  - There will be quizzes on the reading. Why? If you do the readings on time, you will understand the lectures better and do better on the tests and homeworks. The lectures do not recap the readings, they expand on them, adding both broader context and more details. They also update an occasionally out-of-date text. The quizzes are designed to be relatively easy, if you do the reading.
  - The quizzes are on-line via Canvas. You may take a quiz only once, and you have 15 minutes.
Quizzes are associated with lectures. They become available after the previous lecture, and may be taken until 15 minutes before the start of the class the day they are due. For example, there is a reading assignment for Thursday (Chapter 0), so there is an on-line quiz that will appear after today’s (i.e. Tuesday’s) class, and which you must start before 9:15 AM on Thursday.

Quizzes are announced on the class progress page. There is a quiz due this Thursday, but most other quizzes are due before Tuesday lectures.

- Lectures – you are responsible for all the material presented in lectures, some of which is not in the book. (Mandatory)
  - Take notes! Pay attention! ask questions!
  - If you will miss a lecture, make sure someone takes notes for you (you are still responsible for the material)
  - Put your cell phones in silent mode; if you get an emergency call, leave the room before answering. Otherwise, do not use your phone in class.
  - Flashing screens are disturbing to those behind you. Take notes with pen and paper, use a (flat) tablet, or if you use a traditional laptop please sit near the back.

- Recitations
  - Recitations are mandatory
    - Unless otherwise announced, for example during the weeks of midterms and the final
    - Those weeks, recitations are optional.
  - Will be taught by a GTA.
  - Recitations do not recapitulate the lectures. They present tools necessary for software development under unix.
    - Unix, debuggers, profilers, unit test frameworks, etc.
  - A graded exercise is due at the end of most recitations

- Programming assignments – approximately 10 of them (your assignments may vary).
  - They build on each other – the 2nd is to extend the 1st, etc. You can’t skip one.
    - No, I won’t tell you what it coming ahead of time. I want you to learn to write re-usable code.
  - I will tell you what you program should do, but not how to do it. Program design is your task. (Although the lectures may give you hints.)
  - Programming assignments are due when they are due. No credit for late assignments
    - But they build on each other, so if you miss one, you’ll have to do it anyway. You just won’t get credit for it.
  - Assignments are automatically graded: we build test data sets, run your program, compare results to true results.
    - If it doesn’t make and compile, you get 0 points
    - Assignments may have special instructions
      - For example, compiler warnings usually deduct points (around 5%)
      - On some, memory issues detected by valgrind will deduct points
• One assignment will be graded on speed
  ▪ Testing your code is your responsibility. Think of difficult test cases. Try them.
    Never trust the user or a file – make up bad inputs, make sure your program
    doesn’t crash.
  ▪ If your program crashes on an input file, you get no points for that case.
  ▪ We will release the test set so you can double check our tests
    • If you find a discrepancy, talk to the GTAs
    • If you disagree with policy, talk to me
  ▪ What about documentation? Style? We don’t grade for these. We don’t have to.
    The assignments build on each other. After 3 months, documentation and style
    are self-rewarding. Trust me.
• Assignments are compiled and graded ON DEPARTMENT MACHINES. It doesn’t matter if
  it worked on your home laptop...
• My laptop crashed is not an excuse. You have access to department machines, and they
  are backed up nightly. If you choose not to use them, you take responsibility...

• Exams
  o 1st midterm (Tue, Sept 19th) : 10% of your grade
  o 2nd midterm (Tue., Oct. 31st) : 10% of your grade
  o Final exam : 20% of your grade (Mon., Dec 11th)
  o I reserve the right to change the midterm dates with one week’s notice (in case we fall
    behind or get ahead as a class).
  o Warning: I give hard tests. The mean or the raw score is often close to 50%, with a large
    standard deviation. This is how I get a true reading on what you know, and make sure
    that one “stupid mistake” doesn’t torpedo your grade. But it can be demoralizing. Don’t
    give up or get flustered: the scores will be curved.

• Grading
  o Your overall score calculated as:
    ▪ 40% programming assignments
    ▪ 40% tests
      • 10% each midterm
      • 20% final
    ▪ 9% quizzes
    ▪ 11% recitations
  o Tests are curved (they are very hard). Other grading elements may be curved, at my
    discretion.
  o To get a C or better, you need:
    ▪ Overall score of 70% or better
    ▪ At least 65% on programming
    ▪ At least 65% on tests
  o I reserve the right to apply a curve at the end, but may not.

• Resources
Your text (Weiss)
References (Stroustrup or www.cplusplus.com)
The CS253 class web site (www.cs.colostate.edu/~cs253)
  - Important news on home page
  - Progress page provides links to lecture notes
  - Assignments on assignment page
  - Recitations on recitation page
  - Resources page has links to additional resources
Canvas
  - Keep track of your grades
  - Take on-line quizzes

* Academic Integrity
  - I hate this topic. Most of you are honest and won’t have any problems. But...
  - All programming assignments, recitation assignments, quizzes and tests in this course are solo projects. They may not be done in teams. You may not copy code from the internet. You may not submit any work that is not your own. Ever.
  - We will actively look for cheating. No, I will not tell you how.
  - The department and university policies on academic integrity will be strictly enforced, and all cases reported to the university. If you are unclear about these policies, there are links on the class web site (syllabus page).
  - That includes the rule that you are responsible for making sure that no one cheats off you. You are responsible for keeping your own work secure.
  - Does this mean you can’t talk with your classmates? No. It means you can’t write code for them, or they for you. You should never share code, or show someone your code. Abstract discussions are fine.
  - When in doubt, ask me.

* Unforeseeable Emergencies
  - The rule about no late assignments has one exception: unforeseeable emergencies.
  - Examples include: death in family, illness (usually requiring hospitalization), house fires, etc.
  - This probably won’t happen to you ... but there are over 100 of you. Unfortunately, there is a significant likelihood that it could happen to someone in this room. It happened to me as a freshman. If it does happen to you, talk to me. Tests can be rescheduled, assignments delayed, etc., if the emergency justifies it.

Assignment: Read Chapter 0 (yes, 0) of Weiss for Thursday. There is an on-line quiz that closes at 9:15 Thursday morning.