Welcome to the class notes for CS253 (spring semester, 2015). Although it may surprise the more cynical among you, professors are not just “winging it” when we give lectures. Before every class, we write out notes on what we intend to say. This semester, as an experiment, I am handing out these notes to you, the students. There are both pros and cons to this. If used correctly, I believe these notes can help you learn more and do better. Unfortunately, misused they could actually lead to poorer performance. So let me briefly describe these notes and what they are and aren’t good for in this prologue, in the hope that you will use them well. At the end of the semester, we will see if they helped or hurt.

Your primary text for this class is *C++ for Java Programmers* by Mark Allen Weiss. It is important that you read this book. It provides the overall structure for the class, and describes many of the differences between C++ and Java in detail. Moreover, I do not generally repeat the material from the book in lectures. That would be redundant. The lectures emphasize what is important in the text, fill in holes, update the text where it is outdated, and occasionally disagree with the text, but they do not repeat it. If you try to use these notes as a replacement for the text, it will hurt you. Besides, the text really isn’t very long, so the reading assignments shouldn’t be too taxing.

The lectures go beyond the text in many ways. The text, to be blunt, is useful but shallow. The important topics in this class, e.g. memory management, styles of object orientated programming, and how C++ is implemented “under the hood”, are only discussed superficially by Weiss. Weiss instead focuses on syntax and details. The lectures expound on the major topics much more thoroughly. Indeed, we go will through seven weeks where there are no reading assignments in the book; all the material in this part of the course comes from the lectures.

I strongly recommend attending lectures. When attending lecture, I recommend taking notes. Even if you never study your notes, the act of taking notes during class makes you think about and integrate the material in a way that improves learning. Besides, these notes reflect what I intended to say, not what I did say. Often lectures are modified on the fly in response to student questions (or simply because I see that puzzled “I am lost” expression in the audience). This is particularly true in terms of how the lectures relate to the homework. These notes are not a good replacement for notes you take on your own.

What these notes are good for is a study aid before midterms and finals. They should remind you of what was discussed in class, and if you take your own notes you will find that sometimes your notes aren’t clear, and sometimes my notes aren’t clear, but together the information you need is there. If not, they indicate topics you might want to ask about in class or during office hours.
Lecture 1a: Administrivia  
Wednesday, Jan 21\textsuperscript{st}

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): Ankit Biradar and MuthuKutti Raja Selvakumar (Muthu).

Let me start with some quick, procedural notes. This is a MWF class, so the lectures are only 50 minutes long. That means we can’t afford to waste 5 or 10 minutes of every lecture getting started. Lectures will begin at 10:00 sharp. That means you should be in your seat ready to go by 10:00, 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 10:50, 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 well 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 class on Friday, so I have to get to the administrative aspects of the class first. So 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 Friday 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. The quizzes are designed to be relatively easy, if you do the reading.
  - The quizzes are on-line via RamCT. You may take a quiz only once, and you have 15 minutes.
The quizzes become available after the lecture in which they are assigned, 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 Friday (Chapter 0), so there is an on-line quiz that will appear after today’s class, and which you must start before 9:45AM on Friday.

Quizzes are announced on the class progress page. There is a quiz due this Friday, but most other quizzes are due before Monday 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)
  - Lectures begin at 10:00, not 10:05 or later. That means you are at your desks ready to go at 10:00, not walking through the doors of Clark at 10:00. “To be on time is to be late; to be early is to be on time.” Have respect for the class, your peers, and me.
  - I will end class close to 10:50 in return.
  - Put your cell phones in silent mode; if you get an emergency call, leave the room before answering.
  - Flashing screens are disturbing to those behind you. Take notes with pen and paper, use a (flat) tablet, or sit near the back.

- Recitations
  - No recitations this week (MLK day on Monday)
  - Otherwise, 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 an 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 will tell you what you program should do, but not how to do it. That is your task. (Although the lectures may give you hints.)
  - They 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 compile, 0 points
• Assignments may have special instructions
  • On some, compiler warnings deduct points (up to 10%)
  • On some, memory issues detected by valgrind will deduct points
  • Some assignments 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.
  o Assignments are compiled and graded ON DEPARTMENT MACHINES. It doesn’t matter if it worked on your home laptop...
  o 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 (Wed, Feb 24th) : 10% of your grade
  o 2nd midterm (Wed., April 13th) : 10% of your grade
  o Final exam : 20% of your grade (Thurs., May 12th)
  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
    • 10% quizzes
    • 10% recitations
  o Tests are curved (they are very hard). Other things 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
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
  - RamCT
    - Keep track of your grades
    - Take on-line quizzes
    - Links to videos of lectures.

- **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 Friday. There is an on-line quiz that closes at 9:45 Friday morning.