Lecture 01

Introduction

7 Horsebow Moon, Imperial Year 2023

Song of the day: **私たち** by Elaiza Ikeda (2023)

Sections

1. **Character Stats**
2. **How To Succeed In This Course**

Part 1: *Character Stats*

Hey, my name is Sebastián, and my last names are Romero Cruz. I teach about programming for a living.

I have been teaching at Pace since January 2021, making this my seventh semester as an instructor. I got my bachelor of science degree at NYU, in chemical and biomolecular engineering, and my master's in computer science.

Here's how you can reach me if you have any questions:

Email	Office Hours
scruz3@pace.edu	**calendly.com/profromerocruz**

***Figure 1**: Prof. Romero Cruz's contact information.*

Unless you have a super trivial question, **I much prefer you scheduling office hours with me to ask me questions about the course**. Even if we only spend 5 minutes on a Zoom call, it is a lot easier for me to keep track of my calendar this way, and a lot harder for me to lose track of your e-mail.

Anyway, other things I enjoy are:

- **Slice of life anime**: My favourite is Naoko Yamada's ***Liz and the Bluebird***.
- **Reading**: My current favourite is ***Katalin Street*** by Magda Szabó.
- **Playing music**: I play a cute pink **Squier Bass VI** and a **Rickenbacker 4001C64**.

We basically spent the rest of the class talking about the syllabus, so please refer to it in **Classes**.

Part 2: *How to succeed in this course.*

I'm sure that at least some of you have heard rumours about this course—about how it's not an easy one. There's a weird misconception that, just because this course is listed as introductory, it's supposed to be easy. I, honestly, think that it's the other way around. To me, this course is difficult *because* it's your first exposure to computer science, and **computer science is hard**. Anybody who tells you that computer science is easy is either lying, trying to show off their CS121 grade, or is selling you a UDemy course. In all three cases, you should ignore them.

To paraphrase **Fireship**, learning how to program is almost identical to the experience of picking up an instrument. Yes, there are certainly differences in talent between musicians, and yes, there will always be

"prodigies" that can pick up instruments with seemingly very little effort. Similarly, there *are* 10x programmers out there who can pick up computer science very quickly. The reality, though, is that most of us aren't like that. So, just like one would with the piano, oboe, or guitar, we have no choice but to practice in order to become good at it. My recommendations, thus, are as follows:

- **Program at least once a day**: Our homework assignments are, generally speaking, composed of several individual programming problems. If you have five days until the deadline, then split up the work across those five days. The key is that your brain and hands get used to writing programming structures. Computer science is weirdly dependent on muscle memory, so if you program every day, even if it's just a small, simple problem, it'll get easier over time.
- **Do not cram for exams**: This ties into the point above. I understand that, as students taking four+ classes every semester, you don't always get the benefit of time when studying for exams. That being said, if there's a class where cramming will get you nowhere, it's this one. Again, it's literally as if you were only started preparing for a violin recital the night before. Yes, you may be able to derive some very short term benefits (like not failing the exam), but it simply will not stick properly.
- **Get help fast, and early**: This course offers a massive amount of resources outside of lecture, not least of all a veritable army of course assistants. If there is a concept that you're struggling with, ask questions during lab, talk one of our TA during lab, go to the tutoring centre, and/or schedule a meeting with us instructors. Concepts in computer science often build off each other, and this is doubly true for this class. Don't hesitate; ask for help.
- **Memorisation will get you nowhere**: Computer science was literally created in order to solve problems, of which not two are the same. Memorising the solution to problem A from a homework assignment does not necessarily mean that it will help you solve problem B on the exam, even if they are extremely similar. Try to make an effort to understand why each line of code is doing something. If you do, memorising will be almost trivial.

Next: **Programming Fundamentals 1**