How Does the Light Shine?[1]

Robin Dawes

September 7, 2021

[1] With apologies to John Fogerty

Reflects on the folly of relying on technology for something as human as teaching!

.

The Plan

The concept was good. I had a good plan for today's lecture, I had materials prepared, I had recommendations for software for you to install, I had interesting examples, I had exercises for you to work on ... Alas, reality stepped in and delivered a non-functioning AV system (which I realized later almost certainly included the mike I was trying to use) and a failed internet connection[2]. The latter was all the more frustrating because I know that everyone else in the room had good internet connections!

[2] How does the light shine? In the (lecture) hall of **Shambolah**!

The irony of the fact that the Computing professor was the only one who couldn't get connected is not lost on me. It's embarrassing! My only consolation is that the two tech guys couldn't figure out my internet problem either.

The good news is that when I got back to Goodwin Hall I sat down with one of our School of Computing tech guys and got the connection problem sorted out.

Here is a list of what I *intended* to cover in today's class. I'll write about them here instead.

- **Classroom protocols**
- **Welcome**
- **Puzzle**
- **My bio**
- **The goals of this course**
- **Course information**
- **Installing Python**
- **Choosing an IDE**
- **Some Python examples**

Classroom Protocols

WE ACTUALLY COVERED these pretty well. You can find them on this website as well. Nothing more to be said.

Welcome

THIS IS MY CHANCE to welcome you to Queen's and to the School of Computing. I've been here for a long time and I'm very proud of the curriculum we offer, and the strong, supportive community that faculty and students in the School have created.

Puzzle

I LOVE GOOD PUZZLES, particularly when they are relevant to computing. I have a very good one to offer this week, but you will have to come to class on Thursday to find out what it is.

My Bio

YOU GRACIOUSLY ENDURED my stories about plastic computers and other primitive technology. I'll not inflict more of them here.

Goals of the Course

YOU CAN FIND the "learning outcomes" of CISC-121 on the School of Computing webpages. I think there is a much more general goal, which is very hard to quantify: this course should teach you to think like a computer scientist. That involves seeing the world and approaching problems with a particular mindset.

Here's something to ponder: roughly speaking, a physicist sees the world as interactions between energies, forces and masses. A chemist sees the world as interactions between atoms and molecules and energy, and they study the properties and behaviour of substances. A biologist sees the world as interactions between living organisms and the environment they exist in (including other organisms). So how does a computer scientist see the world?

Course Information

YOU CAN FIND information about the course elsewhere on this website. There's no need to repeat it on this page.

Installing Python

MANY COMPUTERS NOW come with Python pre-installed. If yours does not, a Google search on "install Python Windows 10" or "install Python Mac" or "install Python Linux"[3] will give you detailed instructions. A good installer will also install "pip" (or perhaps "pip3") - this is a good thing, which you will need later in the course.

[3] My personal preference.

The current release of Python is 3.9.x but in this course you will be fine with any version of Python 3.

Choosing an IDE

THE ABBREVIATION IDE stands for Integrated Development Environment. In its simplest form, an IDE is a software tool that lets you write, edit and test programs. This is a real step forward from the old days when writing a program and running it were completely separate activities, sometimes even involving different machines.

It's not mandatory to use an IDE in CISC-121, but I strongly recommend it. It will make your life much easier.

Python installations often include the **Idle** IDE. It's free, very popular and pretty easy to learn - there are lots of online instructions for using it.

My favourite IDE is **SCiTE**. It's also free, quite simple and very easy to learn. It does a great job of displaying Python programs (highlighting key words in different colours, etc.) and it works for many other programming languages as well.

If you want to try out a much more sophisticated IDE, take a look at **Eclipse**. It's also free (maybe not for Macs), but be warned: it's a very complex tool, and it is strongly oriented towards Java programming. You can find online instructions for converting Eclipse to a Python IDE but it is a complicated process and I haven't personally tried it.

Some Python Examples

On the page that links to these notes I have posted a few very short Python examples that I intended to work through in class today. For some reason [4] Drupal insists on adding "txt" to the end of the names of Python programs when I upload them. If you plan to download and experiment with these programs you will need to strip the "txt" off the end of the file names.

[4] security-related?