## **Day 1**

Intro and Overview

* Instructor Intro
* Awesome Denoising Example - really sell this one
* Overview of the Course

Programming

* *For the slides part of the lecture, don’t describe syntax much. When working through the notebook, describe the syntax in greater detail.*
* “Programming isn’t…” - this slide is arguably the most important slide of all, really emphasize how being a “great coder” isn’t the most important skill
  + Also, talk about proper use of LLMs for coding assistance
* Debugging
  + Talk about Rubber Ducking - ideally we get rubber ducks to hand out
* Notebook setup - Students need to turn of AI assistance feature in notebook for now - encourage them to use the assistance in the future, but for learning basics it will hurt the learning process
  + Steps: 1. Open Notebook, 2. go to settings in top right, 3. go to AI assistance, 4. click the “Hide generative AI features” button
* Print() function
  + For fun, mention that Python got its name because the creator was a fan of Monty Python - fitting that there is a Holy Grail
* Recap math and logic- at the recap slides, jump over to the notebook for the part that the recap slide says too
  + The notebook is made to complement the ideas in the slides, and should be filled out periodically with each section covering something from the recap bit
* Variables - make a point of good naming practice
* Recap dtypes etc. - go to notebook
* Accessing Items - note on how indices start at 0
* 2D lists
  + Mention that you can have 3D lists, list of 2D lists; 4D lists, list of 3D lists, etc.
* Recap loops and lists - go to notebook
* Recap conditionals - go to notebook
* Classes - they won’t be coding any classes, but still describe the general outline of a class
* Packages
* Recap - after the recap, go to the next slide which is Fibonacci Project
* Fibonacci Project - in the notebook they are tasked with writing a function that gives fibonacci numbers (probably will be done with loops instead of recursion). This will work best if you ask students to solve it with the class. Ex. “What would be the first step?” someone responds “make some variables”, etc.
* 10 min break

Mini Projects

* Talk about how to approach coding problems - writing pseudocode
* Then go to the mini-projects in the notebook - the point of the mini-projects is that they do everything by themselves. Of course, answer any questions they have, but answer them in a way that guides them to figuring it out for themselves.
* These are somewhat difficult problems, so there will be a lot of time allotted for them.
* Afterwards, walkthrough the solutions for the mini-projects - there is a separate notebook for this

Intro to Vectors

* Watch the 3Blue1Brown video that is linked

Homework

* The students will be given a homework notebook. Their job is to fill in the missing code in a random number guessing game.