CS210 Discussion

Week 1

UMass Boston

Welcome to Discussion

- The goal is hands on practice with key concepts
- Interactive, somewhat like a lab
- Bring a laptop

Discussion Structure

- Attendance - iClicker
- Introduction
  - Review
  - Examples
  - Setup
- Breakout
  - Work on the Exercises (collaboration is okay)
  - Ask me or the CA questions
- Show us passing Gradescope tests
  - Leave early

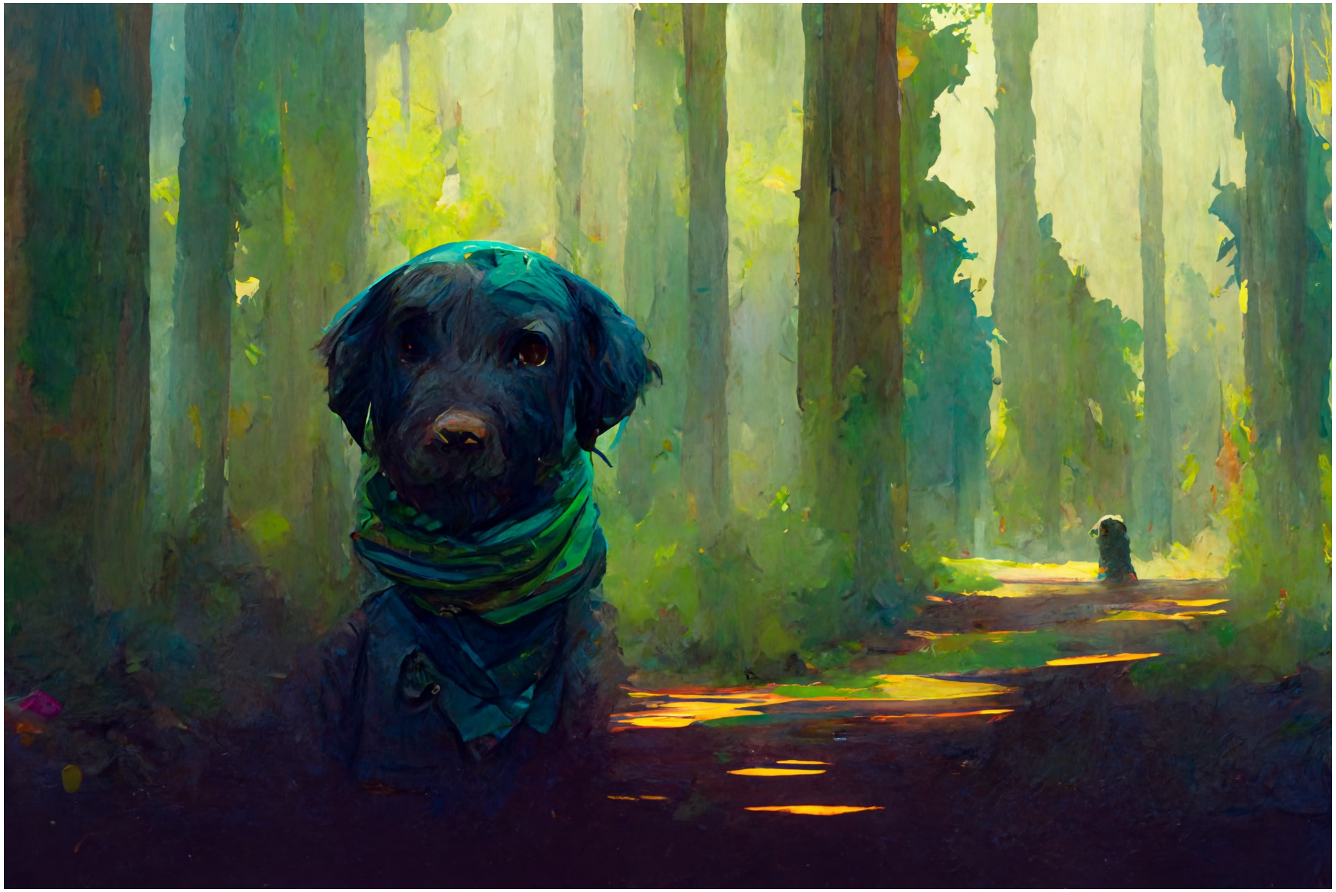
Discussion

- You'll need to finish exercises on your own if not in discussion
- Projects must be done independently
- Office hours are posted on the course website
  - Let us know if the times don't work for you

Tips for Success

- ★ Ask questions early and often
  - The pace is quick and the topics build on each other
- ★ Start the projects as soon as they come out
  - Even if you haven’t been taught the material yet
  - Write down your questions
- Practice patience
- Get comfortable with the CLI
- Try to write easy to understand code and revise as necessary
  - Writing code is like any other writing

Questions?



Today

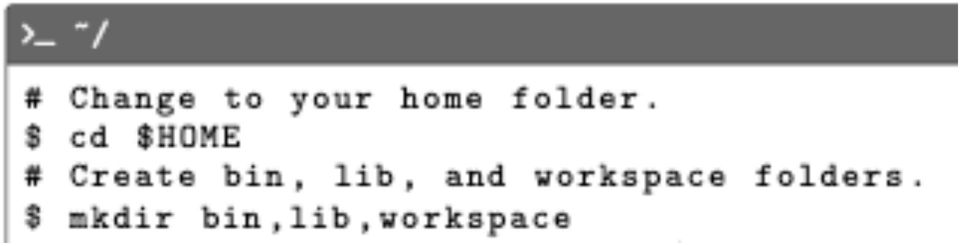
- Intro material
- Setup programming environment
- Finish through exercise 3
    - Show us passing Gradescope tests to leave early

Programming Environment

- Find the instructions on the course website
  - First bullet on the “Resources” page
- Go step by step
- Don’t copy and paste the commands
  - It adds spaces that shouldn’t be there

Comment (don’t type) →

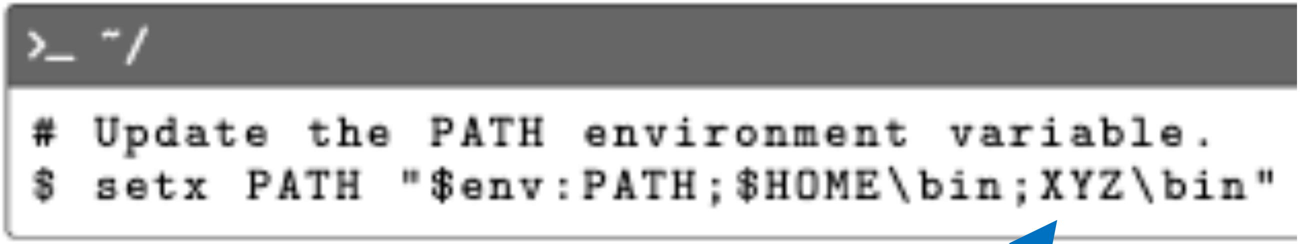
Command →



```
# Change to your home folder.
$ cd $HOME
# Create bin, lib, and workspace folders.
$ mkdir bin,lib,workspace
```

Programming Environment

- Windows common issue
  - Keep in mind where you download and put Java



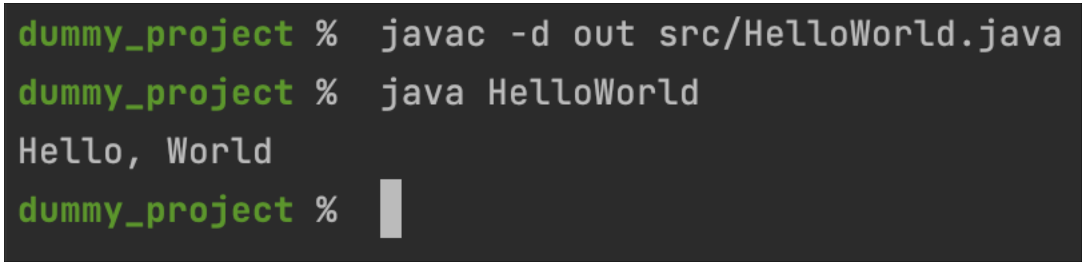
```
# Update the PATH environment variable.
$ setx PATH "$env:PATH;$HOME\bin;XYZ\bin"
```

XYZ should be the path to the
Java folder (wherever you put it)

- I recommend just putting the java folder in your home directory
  - $HOME\jdk-17\bin
  - Doesn't have to be this, just make sure it's accurate to where you put Java

Programming

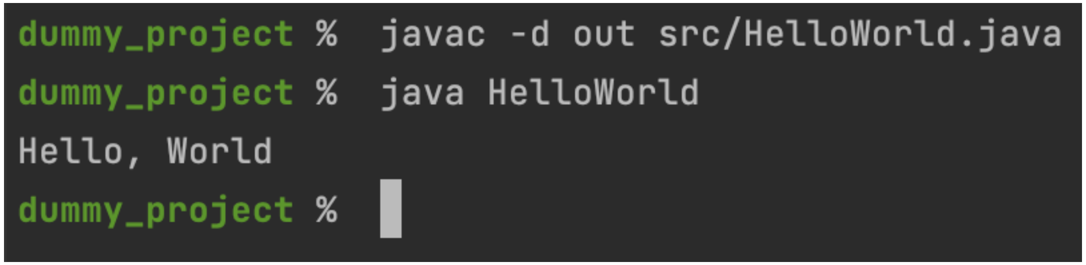
- Recommended workflow
- Write some code → Compile → Run locally to test
- Do this several times while working on an exercise or project
- After the code produces the expected output, upload to Gradescope and see if it passes all tests
  - You can also upload early to get feedback/hints from Gradescope



```
dummy_project %  javac -d out src/HelloWorld.java
dummy_project %  java HelloWorld
Hello, World
dummy_project %  █
```

Programming

- These commands should be run from the assignment folder
  - Use the ‘cd’ command to change directories
- javac ← To compile the code
- java ← To run the code
- If you make changes to the code, you must recompile



```
dummy_project %  javac -d out src/HelloWorld.java
dummy_project %  java HelloWorld
Hello, World
dummy_project %  ▌
```

Exercises

- Getting used to working in Java
- Datatypes
- Arrays
- Input/Output
- Object oriented programming

Exercises / Programming Env

- Work on setting up the programming environment then start the exercises
  - We'll be around to help and answer questions

Hints:

- The `args` array holds your command line arguments, starting at `0`. So `args[0]` is the first argument to your program on the cmd line
- Strings can be parsed into other data types using certain functions/methods
  - For example, to parse a string to a double use `Double.parseDouble(…)`
- Use the Math library to find math functions
  - For example, `Math.sin(…)`
- Use StdOut to print to the command line
  - For example, `StdOut.println(…)`

Questions?

Exercise 1 – Great Circle

All together now