OIM3690 - Web Technologies

Agenda

- Introducing yourself
- Introduction to the course
  - Syllaubs
  - Term Project (*mentioned*)
  - Get familiar with software
- How does the web work?

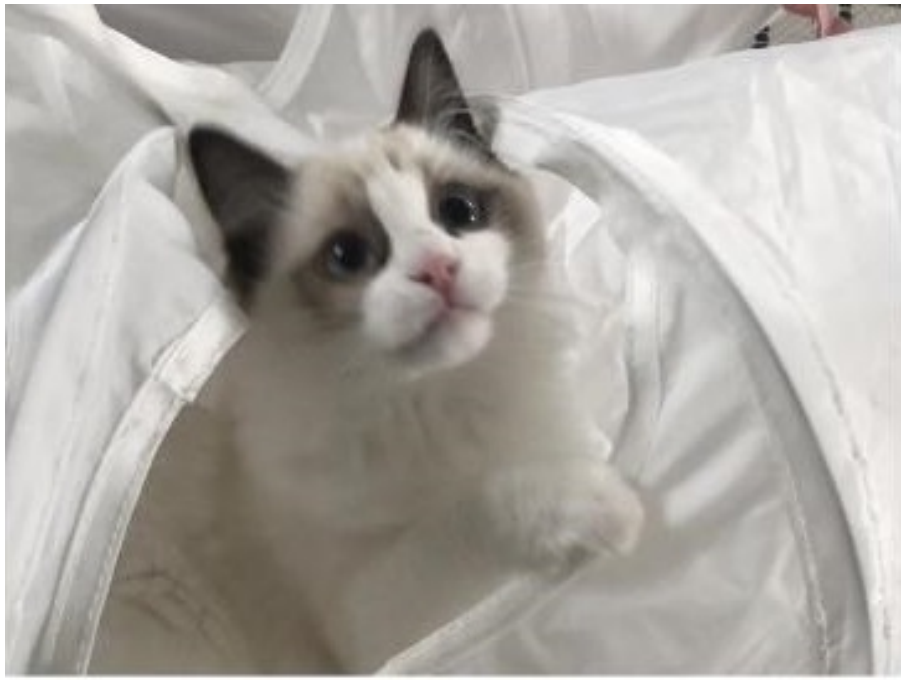
Welcome! How are you doing?

Please introduce yourself, including:

- **Who** are you? **Where** did you come from?
- Are you a **sophomore**, **junior** or **senior**? What was your **best Babson moment**?
- What is your **concentration**? Are you going to rule the world with that?
- **Why** did you decide to take this class?
- Do you have any **programming experience**?
- How was your **2022** and your **winter break**? Did you do **anything exciting** or just binge-watch a lot of Netflix?
- How can we **remember** you?

About Me

- Instructor: **Zhi Li** (李直)
- Email: zli@babson.edu
- Office: Babson Hall 216D
- Office Hours:
  - In-person:
    - Tuesday: 11:30AM - 12:30PM
    - Thursday: 6:30PM - 7:30PM
  - Online via Webex: by appointment



A Quick Survey

1. Have you viewed **source code** of any web page?
2. Have you used **Git/GitHub** before?
3. Have you created **website(s)** before?
4. Have you heard of ***Web3***?

What is this course about?

Well, let me first tell you that this course is ***NOT*** about...

- Web3
- UI design tools (Figma/Sketch)
- Back-end/fullstack/app development
- Web analytics/SEO
- React/Angular/Vue

Seriously, what is this course about?

- How the Web Works
- HTML5
- Semantic HTML
- CSS3 Essentials
- CSS Layout
- Flexbox & Grid
- Responsive Design
- Using GitHub
- Deployment
- Bootstrap/Tailwind CSS
- JavaScript Basics
- Modern JavaScript Syntax
- DOM Manipulation
- Image Processing
- Web APIs
- JSON data
- Geocoding & Maps
- Web Game Development
- Local Storage
- ...

What really matters are ...

- Familiarizing yourself with basic programming concepts and front-end technologies
- Building website from the initial design phase to deployment
- Thinking like a software engineer and a computer scientist
- Learning how to learn and how to get "unstuck"
- Collaborating effectively with engineers and other team members through the use of tools and clear communication
- Equipping you with the tools and mindset to succeed after completing this course

Syllabus

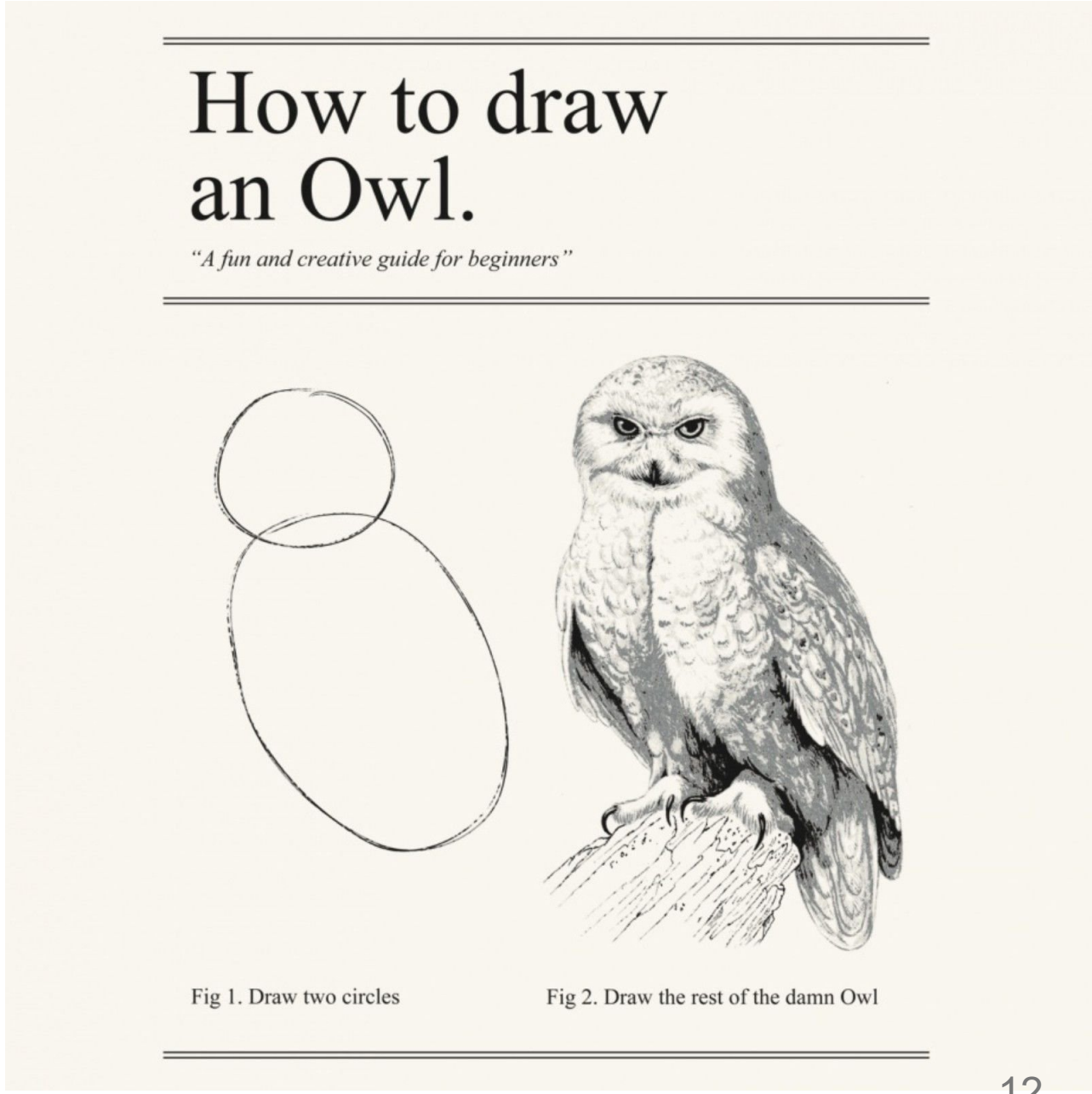
- Course Objectives
- Prerequisites and Textbook
- Software (next slide)
- Term Project (a personal website)
- Exercises/Quizzes/Graded Homework/~~Exam~~
- Grading
- Course Policies

Software

- Visual Studio Code (VSCode), and extensions
  - Live Server
  - Prettier
  - vscode-icons
  - ...
- GitHub Desktop
  - Sign up for GitHub (using Babson email)
  - Sign up for GitHub Student Developer Pack

How to Learn Programming

Programming is hard.



DO NOT take the “couch potato” approach

Practice!

Practice!

Practice!

DO NOT copy and paste!

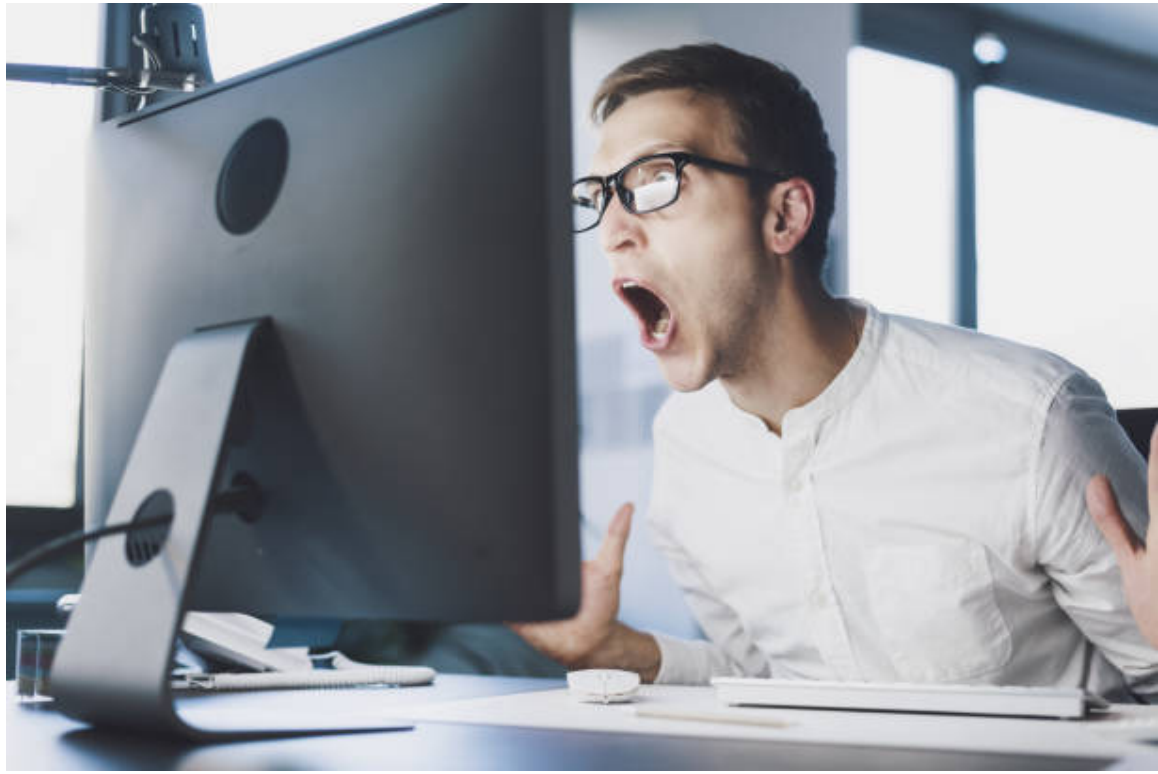
Ask Questions

Ask Questions the Smart Way

- The XY Problem
- How do I ask a good question
- How To Ask Questions The Smart Way
- Getting Answers

DO NOT panic!

- Almost everyone hits a rough patch in the course at some point.
- Don't let it discourage you.
- It's normal!



What if I got "stuck"

- Take a break
- Break the problem down
- Keep trying
- Debug
- Ask for help


Google
stackoverflow
reddit
slack

How to Cheat without Being Caught

- If you're going to cheat, here are some tips:
  - Do not submit code that has a matching md5sum as your friend's code.
  - Don't share your code with others if it is supposed to be individual work.
  - Avoid simply changing comments and spacing, as the code can be tokenized to eliminate these differences.
  - Changing variable names, moving definitions, or copying only part of other's code will be detected as well.
  - You may not use code found on the internet or written by AI.
- If you still decide to cheat, the **only way to cheat safely** is to rewrite the assignment from scratch.

How does the Web work?

- Reading: How the Web works
- Watching: How does the INTERNET work?


ANY
QUESTIONS
?