Rethinking Computing Education with Vocareum and Canvas

Prof. Alexander (Alex) Dowling

adowling@nd.edu dowlinglab.nd.edu

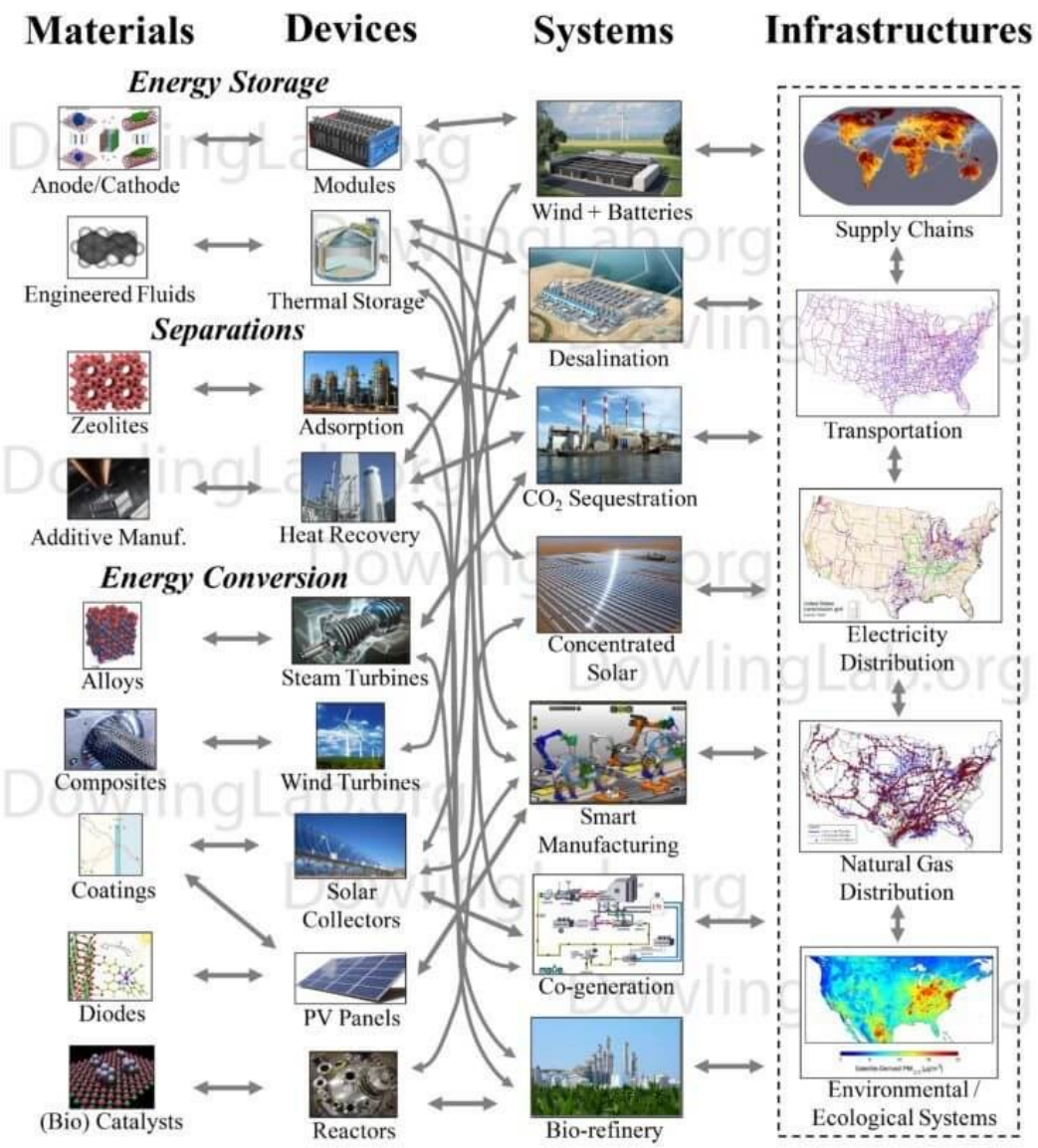
Department of Chemical and Biomolecular Engineering

University of Notre Dame

November 18, 2021

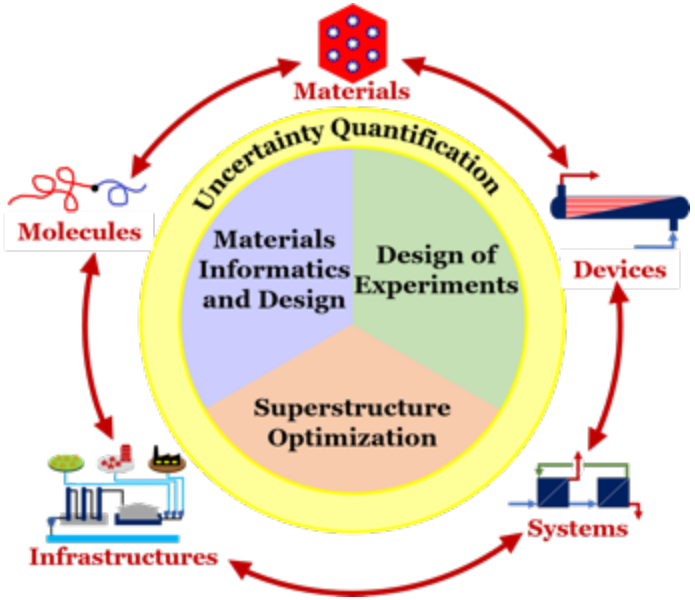
These slides are available at **http://psecommunity.org/LAPSE:2021.0803**

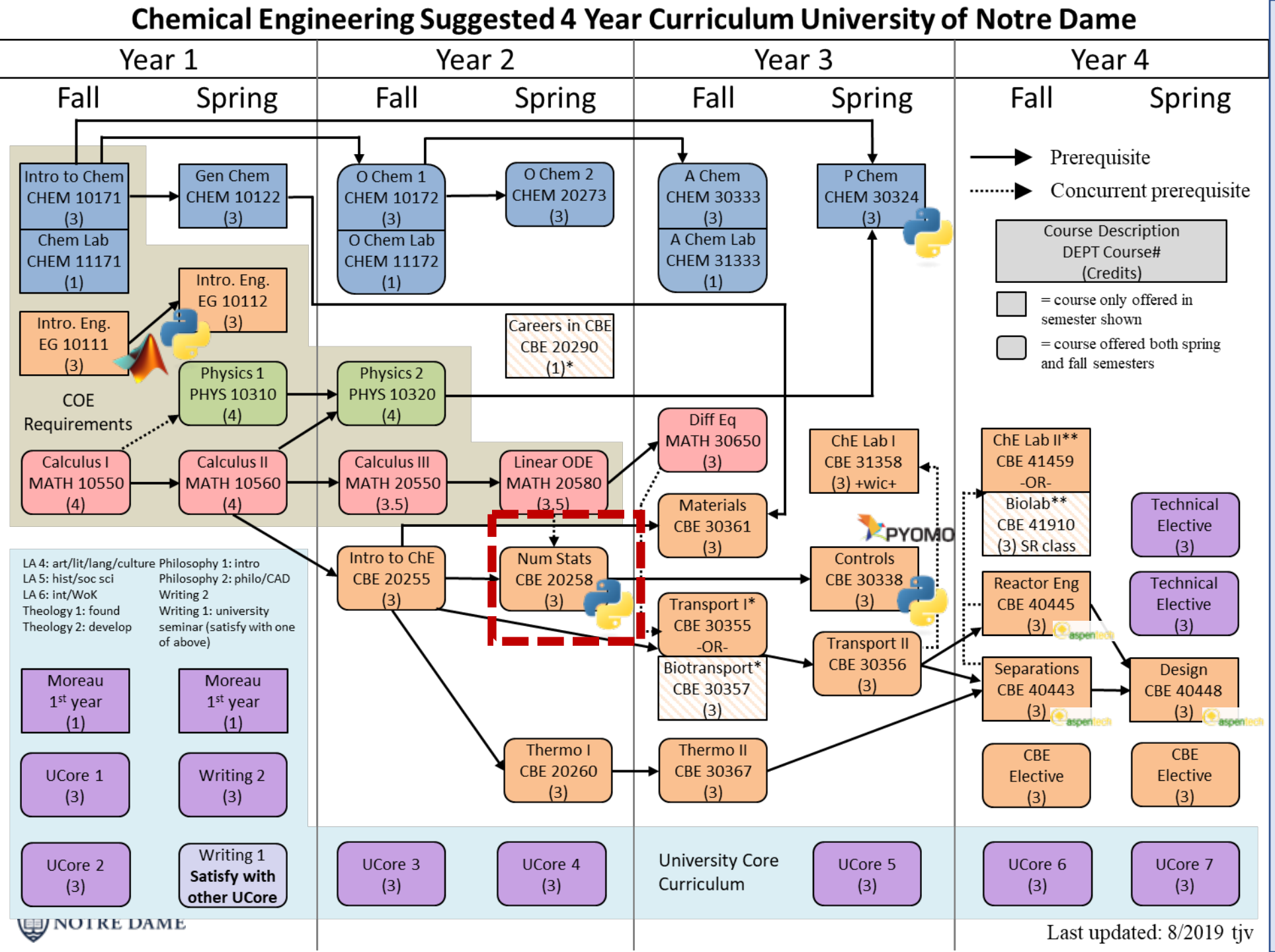
Research: Process Systems Engineering



Themes

- Mathematical Modeling
- Computational Optimization
- Applied Statistics and Uncertainty Quantification
- Energy, Sustainability, & Systems Biology Applications





Current Practice: Computing & Statistics

MATLAB in freshman engineering sequence

Sophomore-required **Numerical & Statistical Analysis** (NSA)

Ad-hoc computing & statistics in upper-level classes:

"You learned this as sophomores… just figure it out" – Prof. Anonymous

Vision

Vertically integrate computing and statistics throughout the undergraduate curriculum

Modernizing Numerical and Statistical Analysis

Backward Course Design

Set Clear Learning Objectives

At the end of the semester, you should be able to…

1. **Create mathematical models** and **apply computational methods** to analyze systems using basic principles of chemical engineering (e.g., mass and energy balances, thermodynamic equilibrium, etc.)

2. **Analyze data** and **quantify uncertainty** using standard statistical techniques and mathematical models grounded in engineering fundamentals

3. Independently plan, implement, and debug short (100 to 300 lines) **Python computer programs** to analyze data, solve engineering mathematical models, and visualize results

Major Changes

Reorganized class topics

- Removed advanced topics (QR factorization, compression with SVD, trust regions, BVPs, PDEs)
- Emphasized fundamentals, especially probability & statistics
- Added mass and energy balance examples

Switched to **Python**, with great student buy-in

Incorporated **active learning** into lectures

Shortened assignments

Active Learning is Essential for Computing and Statistics

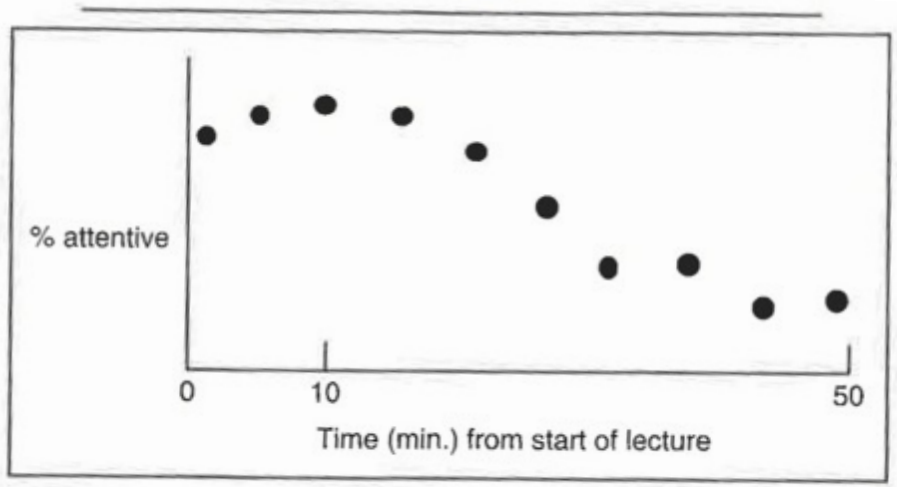


Figure 6.3–1: Attentiveness versus Time in Lecture—No Activities

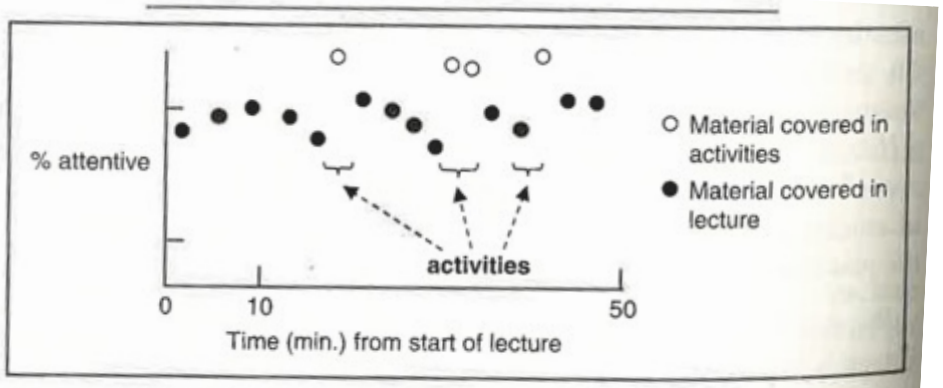


Figure 6.3–2: Attentiveness versus Time in Lecture—Activities Interspersed

Right: Felder and Brent (2015), *Teaching and Learning STEM: A Practical Guide*

Spring 2019: Cloud-based Google Colaboratory (Jupyter Notebooks)

`colab.research.google.com`

Benefits of Google Colaboratry:

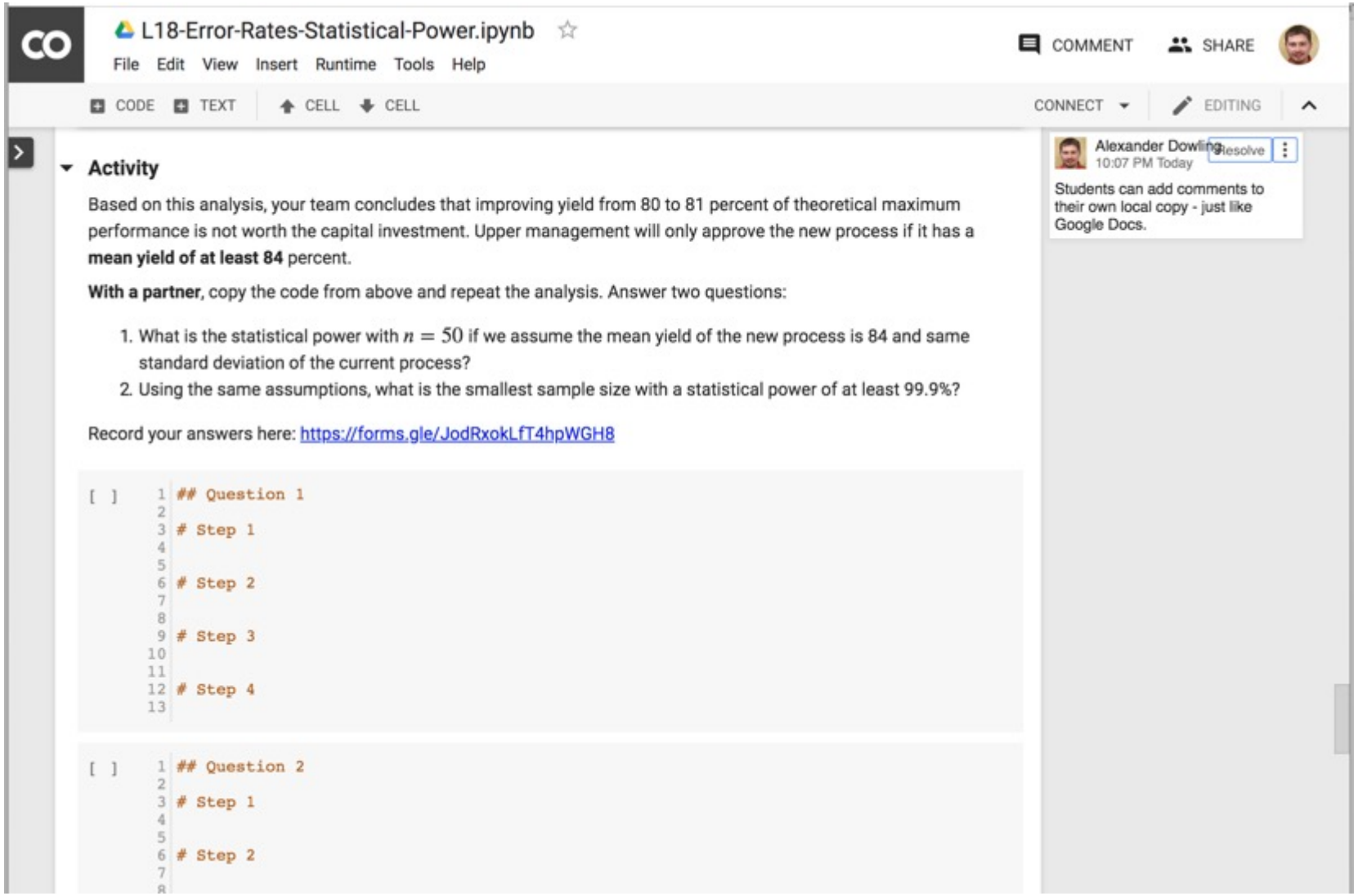
Like Google Docs, but for code

Integrated with **Google Drive**: automatic versioning, easy sharing

Removes barriers to access: students can complete assignments from **any internet connect computer** – no need to support 80+ local Python installations

Facilitates **active learning**

Free



I show this slide on day 1.

Making your time more effective

Easy — Complexity of Concepts and Applications — **Difficult**

Traditional Class (plan)

Lecture	+	Tutorial	+	Studying	+	Homework	=	Total
3 hr / wk		1 hr / wk		2 hr / wk		3 hr / wk		9 hr / wk
Initial exposure		*Practice*		*But how?*		*Builds on lectures*		

Traditional Class (reality for many)

Lecture	+	Tutorial	+	Studying	+	Homework	=	Total
3 hr / wk		1 hr / wk		1 hr / wk		10+ hr / wk		15+ hr / wk
Difficult to focus 100% of time		*Too busy to start homework before*		*Either skipped or not effective*		*Really difficult if not comfortable with lecture examples*		

Making your time more effective

I show this slide on day 1.

Easy — Complexity of Concepts and Applications — **Difficult**

**We'll start some homework problems during class.*

This Semester

Class Preparation
2 hr / wk

Initial exposure at home

+

Tutorial
1 hr / wk

Practice & jump-start homework

+

Studying
1 hr / wk

I'll teach you how to do this & give extra practice problems

+

Class*
3 hr / wk

Problem solving together

Homework*
2 - 4 hr / wk

Easy extensions of home and class activities

=

Total
9 - 11 hr / wk

This is 100% on task time… i.e., Facebook closed, not watching Netflix, not texting

Fall 2019 - today: Cloud-based Vocareum (Jupyter Notebooks)

www.vocareum.com

Benefits of Vocareum:

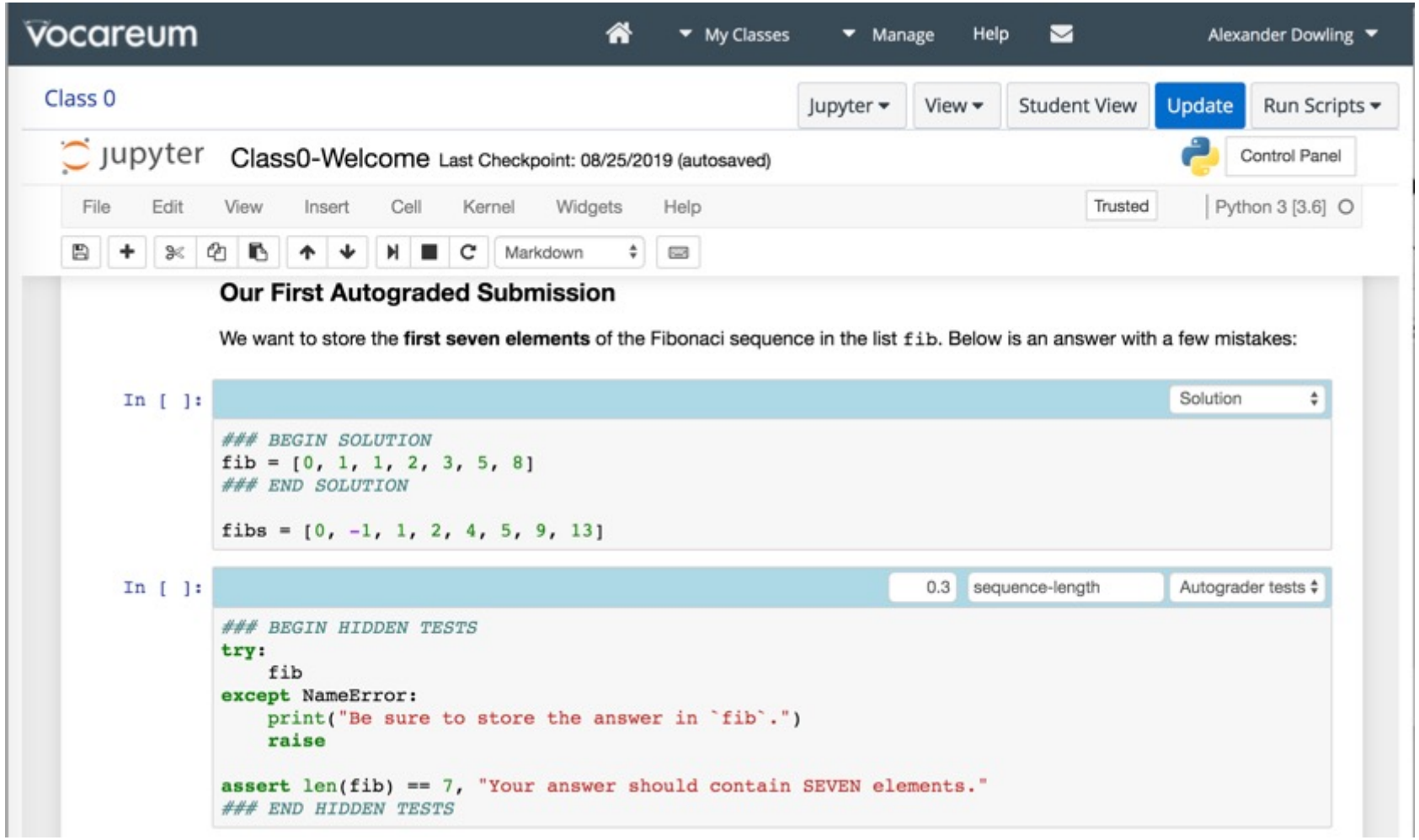
Many of the same cloud-based benefits as Colaboratory

Integrated with **Learning Management System** (e.g., Canvas) and gradebook

Supports **autograding** via nbgrader (with some enhancements)

Supports plagiarism detections (if you want it)

Paid service, but responsive technical support



Fall 2019 - today: Cloud-based Vocareum (Jupyter Notebooks)

www.vocareum.com

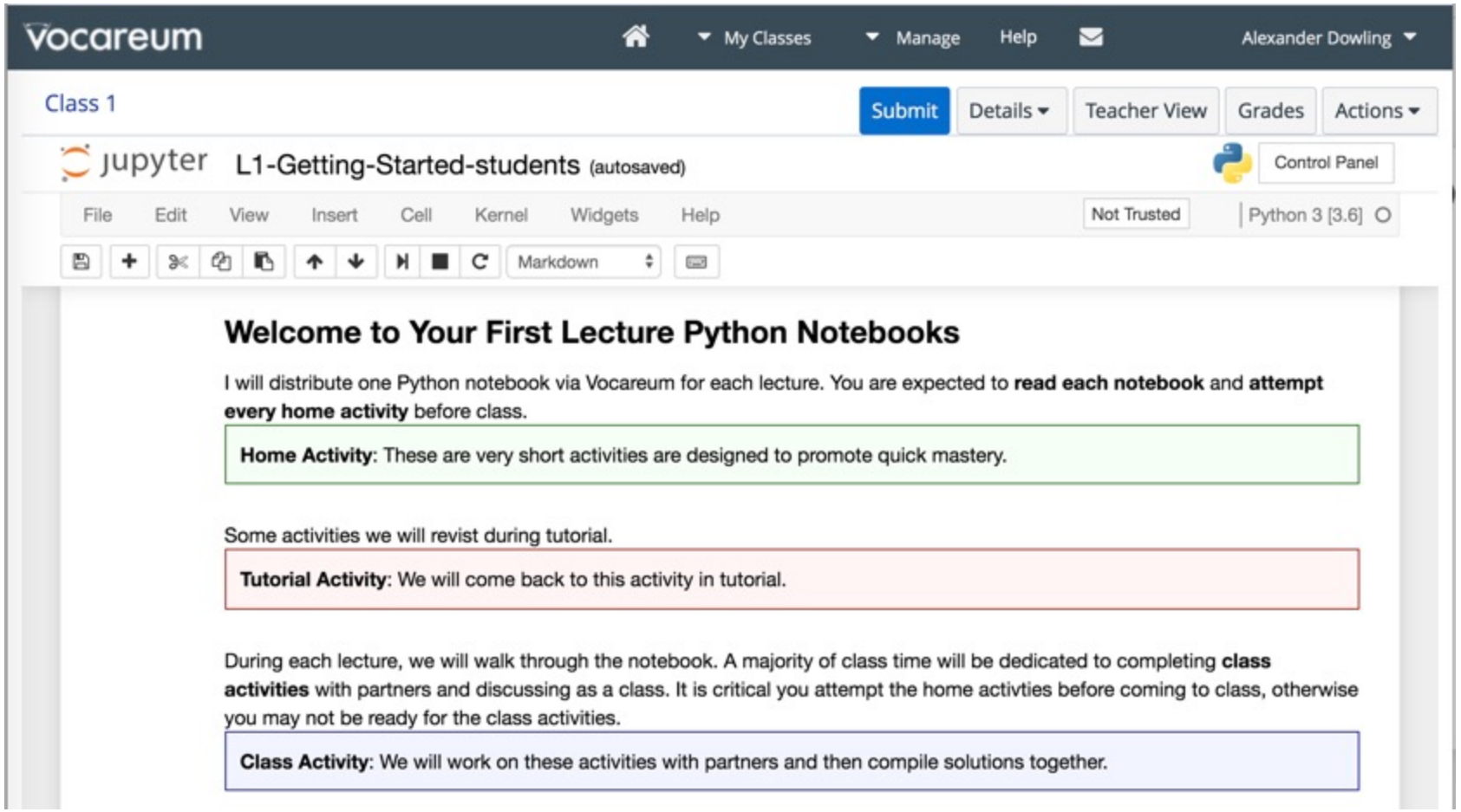
Benefits of Vocareum:

Many of the same cloud-based benefits as Colaboratory

Integrated with **Learning Management System** (e.g., Canvas) and gradebook

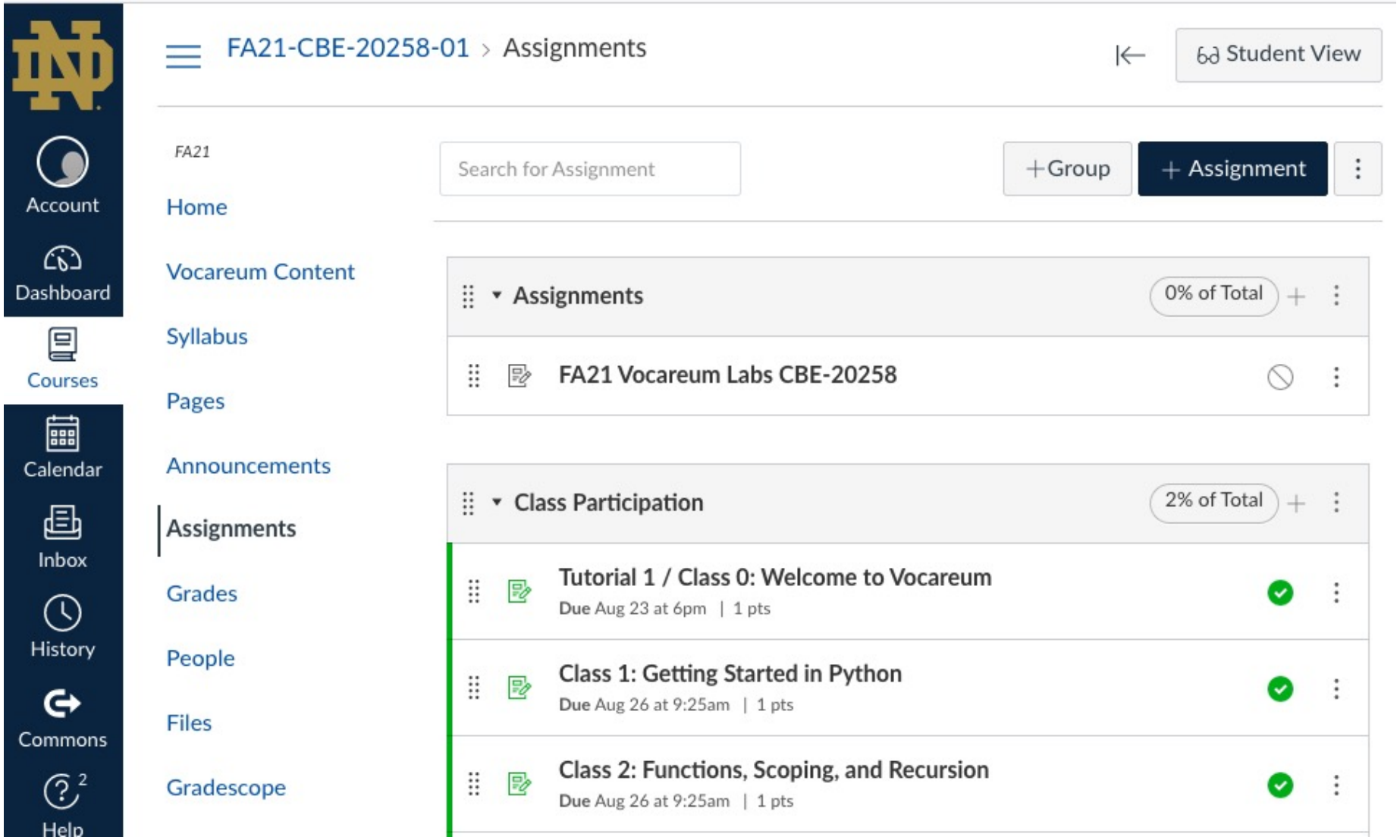
Supports **autograding** via nbgrader (with some enhancements)

Supports plagiarism detections (if you want it)

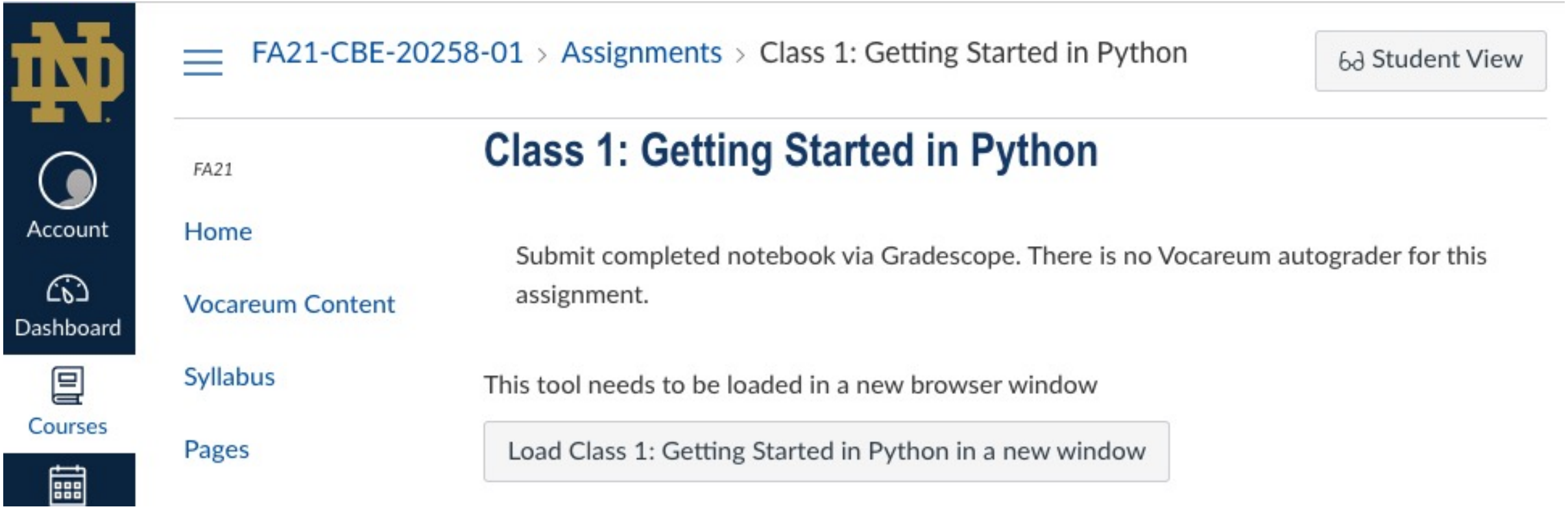


Bottom Line: Autograder (Vocareum) enables accountability for meaningful home activities before class, which translates to more engaging class sessions.

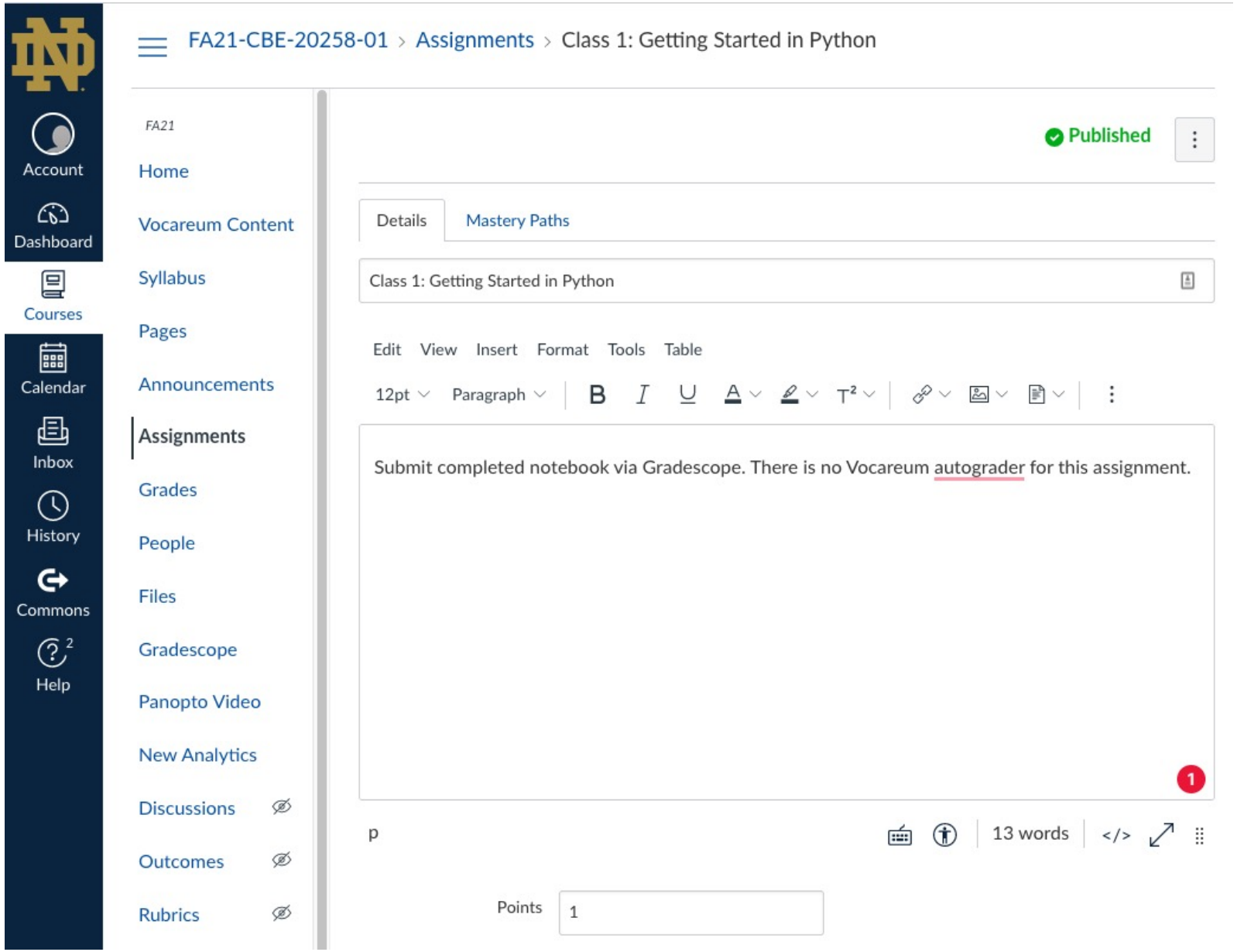
Canvas *Assignments* give students landing page



Each Canvas *Assignment* includes a link to Vocareum



Instructor Manually Creates Each *Assignment* in Canvas

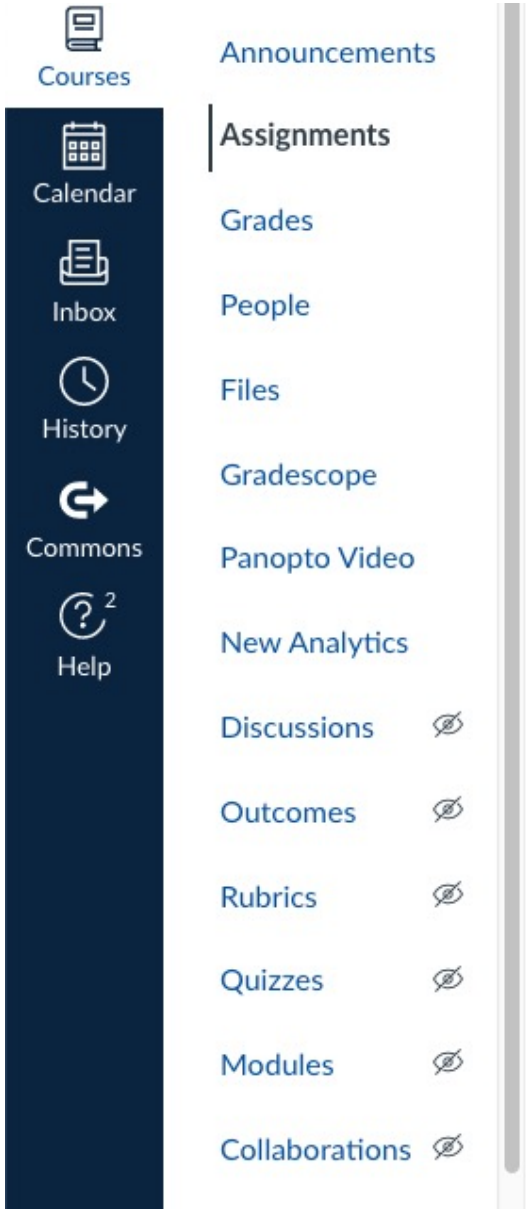


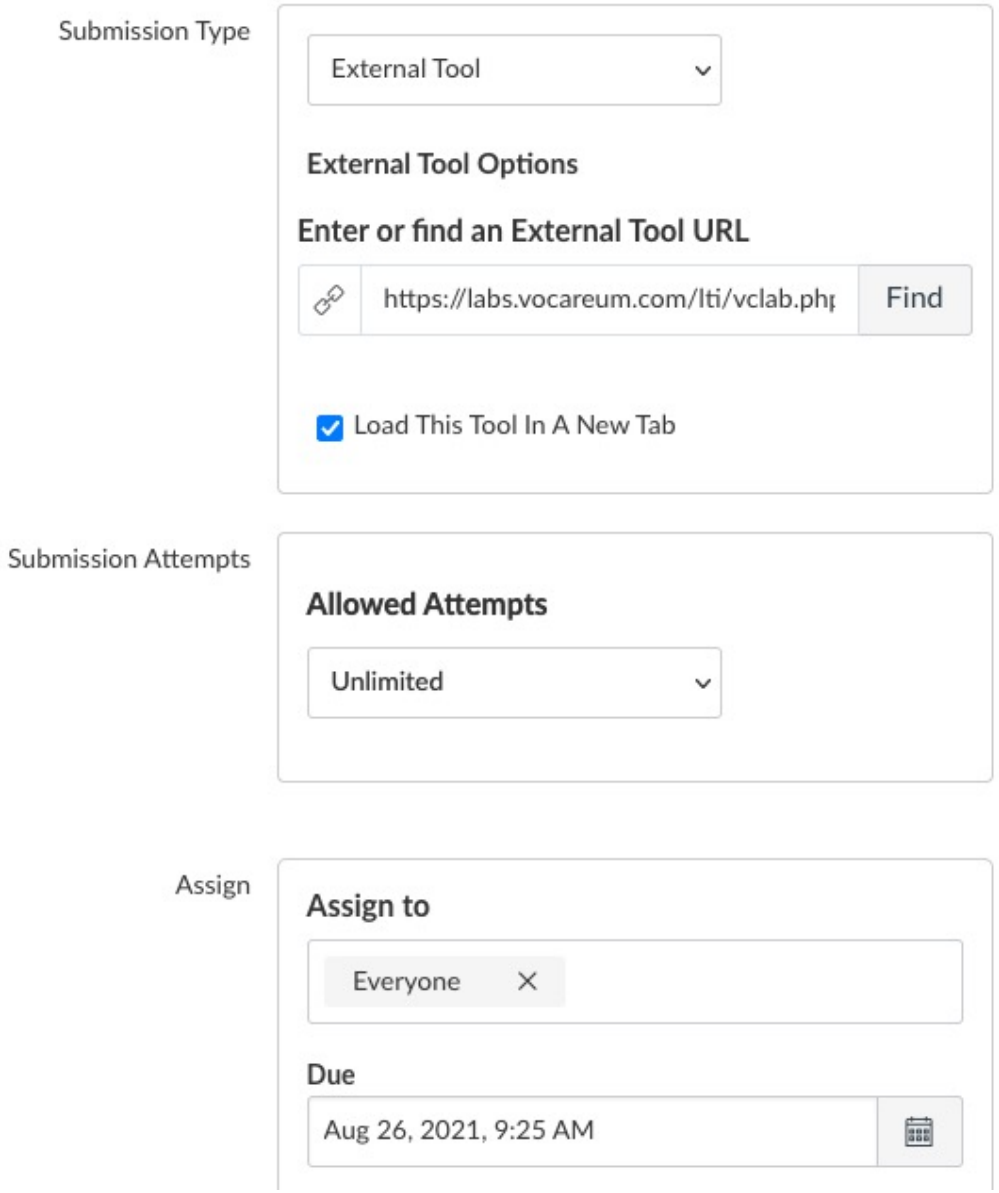
Change title here

Write instructions here

Points in Canvas and Vocareum must match, otherwise grades will not transfer

Instructor Manually Creates Each *Assignment* in Canvas





Select “External Tool”

Copy this URL from Vocareum (easy)

Check this box

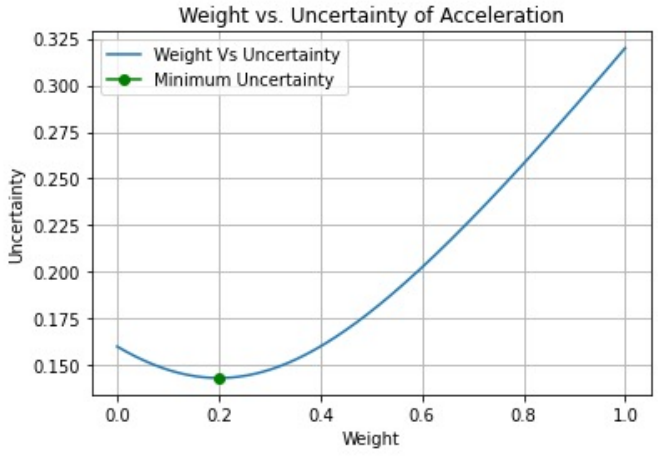
Set due date here

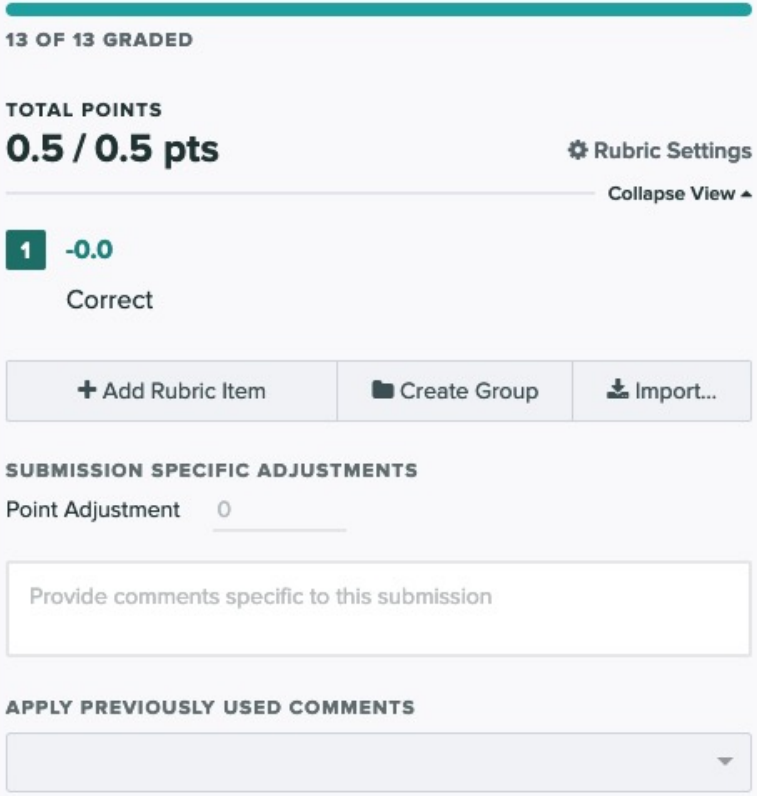
Vocareum + Gradescope for Jupyter Notebooks

Typical assignments require three submissions:

1. Vocareum (autograder)
2. **Gradescope Notebook**
3. Gradescope Written

```
    U[i] = math.sqrt((w[i]**2)*(U_A1**2) + ((1-
w[i])**2)*(U_A2**2))

#plot w vs uncertainity
plt.plot(w,U, label = 'Weight Vs Uncertainty')
plt.plot(w[np.argmin(U)],np.min(U), marker = 'o',
color = 'green', label = 'Minimum Uncertainty')
plt.xlabel('Weight')
plt.ylabel('Uncertainty')
plt.title('Weight vs. Uncertainty of Acceleration')
plt.legend()
plt.grid(True)
plt.show

# Find the minimum uncertainty
index = np.argmin(U)
weight = w[index]

#Calculate Acceleration
A3 = round(weight*A1 + (1-weight)*A2,2)

#Calculate Uncertainty
U_A3 = round(math.sqrt((weight**2)*(U_A1**2) + ((1-
weight)**2)*(U_A2**2)),2)
```





nbpages + Google Colab

CBE60499

Nonlinear and Stochastic Optimization. https://ndcbe.github.io/CBE60499/

View the Project on GitHub ndcbe/CBE60499

CBE60499

This notebook contains material from CBE60499; content is available on Github.

< 1.1 60 Minutes to Pyomo: An Energy Storage Model Predictive Control Example | Contents | Tag Index | 2.0 Optimization Modeling with Applications >

Open in Colab | Github | Download

In [ ]:

```
# IMPORT DATA FILES USED BY THIS NOTEBOOK
import os,  requests

file_links = [("data/Prices_DAM_ALTA2G_7_B1.csv", "https://ndcbe.github.io/CBE

# This cell has been added by nbpages. Run this cell to download data files re

for filepath, fileurl in file_links:
    stem, filename = os.path.split(filepath)
    if stem:
        if not os.path.exists(stem):
            os.mkdir(stem)
    if not os.path.isfile(filepath):
        with open(filepath, 'wb') as f:
            response = requests.get(fileurl)
            f.write(response.content)
```

1.2 Pyomo Mini-Project: Receding Horizon Stochastic Control

Deadline: Friday, March 5, 2021

1.2.1 Assignment Goals

Which platform?

Vocareum

Closed ecosystem, requires authentication (e.g., Canvas)
- More effort for students to access in future
+ Easy to control access

+ **Autograder** with Canvas integration

- More effort to setup/manage

+ Responsive tech support

+ Easy to access all versions of student submissions

Great for computing focused undergraduate classes

colab

Sharing via Google Drive or website (nbpages)
+ Easy to disseminate
- Limited control over access

- Only manual grading via Gradescope

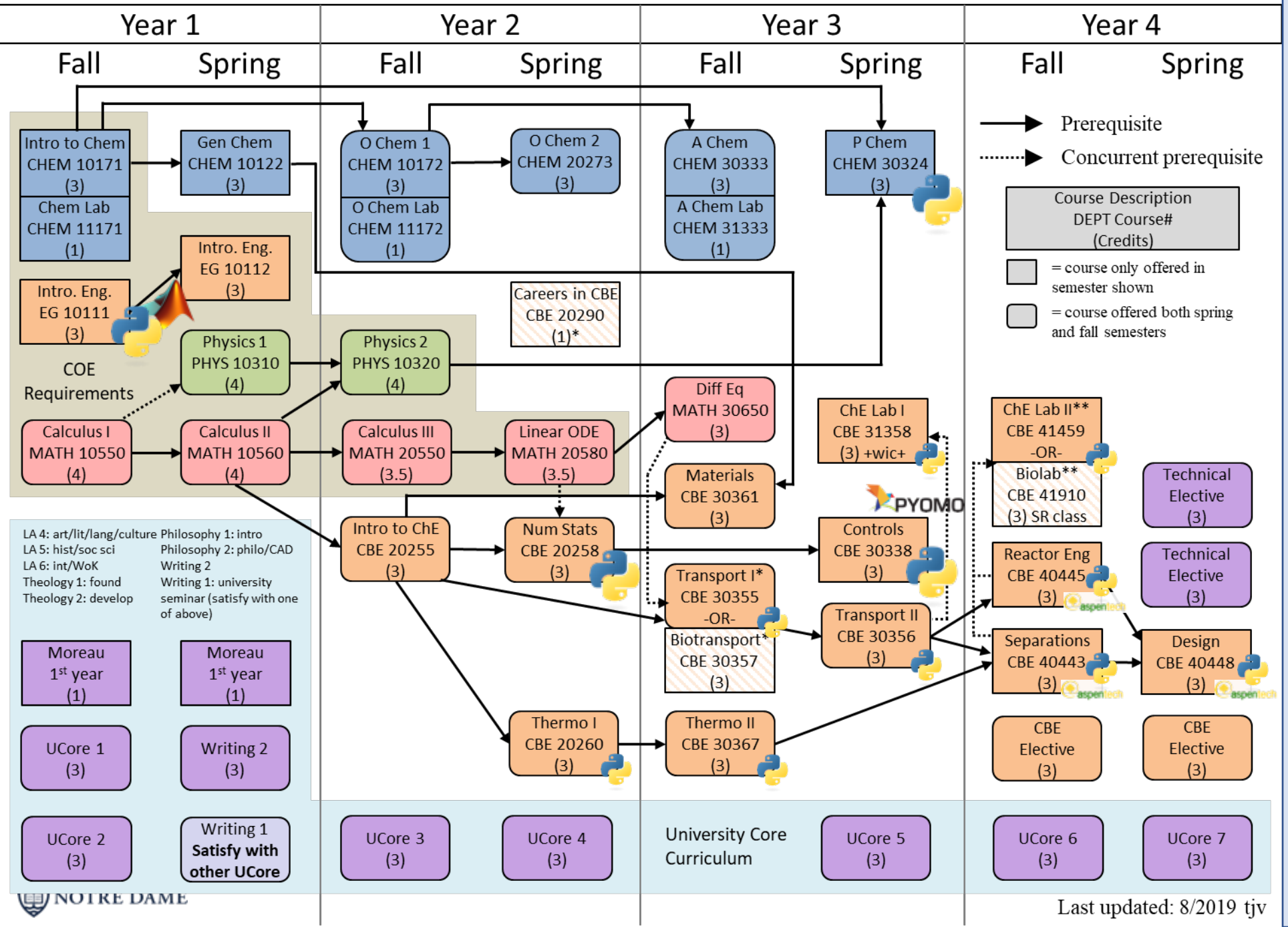
+ Easier to setup

- On your own, fingers crossed Google does not end support for Colab ;)

+ Google Drive automatically stores ~100 versions history for students

Great for graduate classes and occasional class assignments/examples

Chemical Engineering Suggested 4 Year Curriculum University of Notre Dame



Vision

Vertically integrate computing and statistics throughout the undergraduate curriculum

Library of Cloud-based Jupyter Notebooks

Complement existing core CBE classes with examples that use computing and statistics for problem solving

colab

vocareum

Special Thanks

https://github.com/jckantor
Chemical Process Control
Introduction to Chemical Engineering Analysis
Introduction to Operations Research
Process Operations

Prof. Jeff Kantor

Prof. Yamil Colón

Vocareum Pilot

Pat Miller

Xiaojing Duan

Kaneb Center

Kevin Barry

Dan Hubert

Kristi Rudenga

Rethinking Computing Education with Vocareum and Canvas

Prof. Alexander (Alex) Dowling

adowling@nd.edu dowlinglab.nd.edu

Department of Chemical and Biomolecular Engineering

University of Notre Dame

November 18, 2021

These slides are available at **http://psecommunity.org/LAPSE:2021.0803**