Rethinking Computing Education with Vocareum and Canvas

Prof. Alexander Dowling

adowling@nd.edu **dowlinglab.nd.edu**

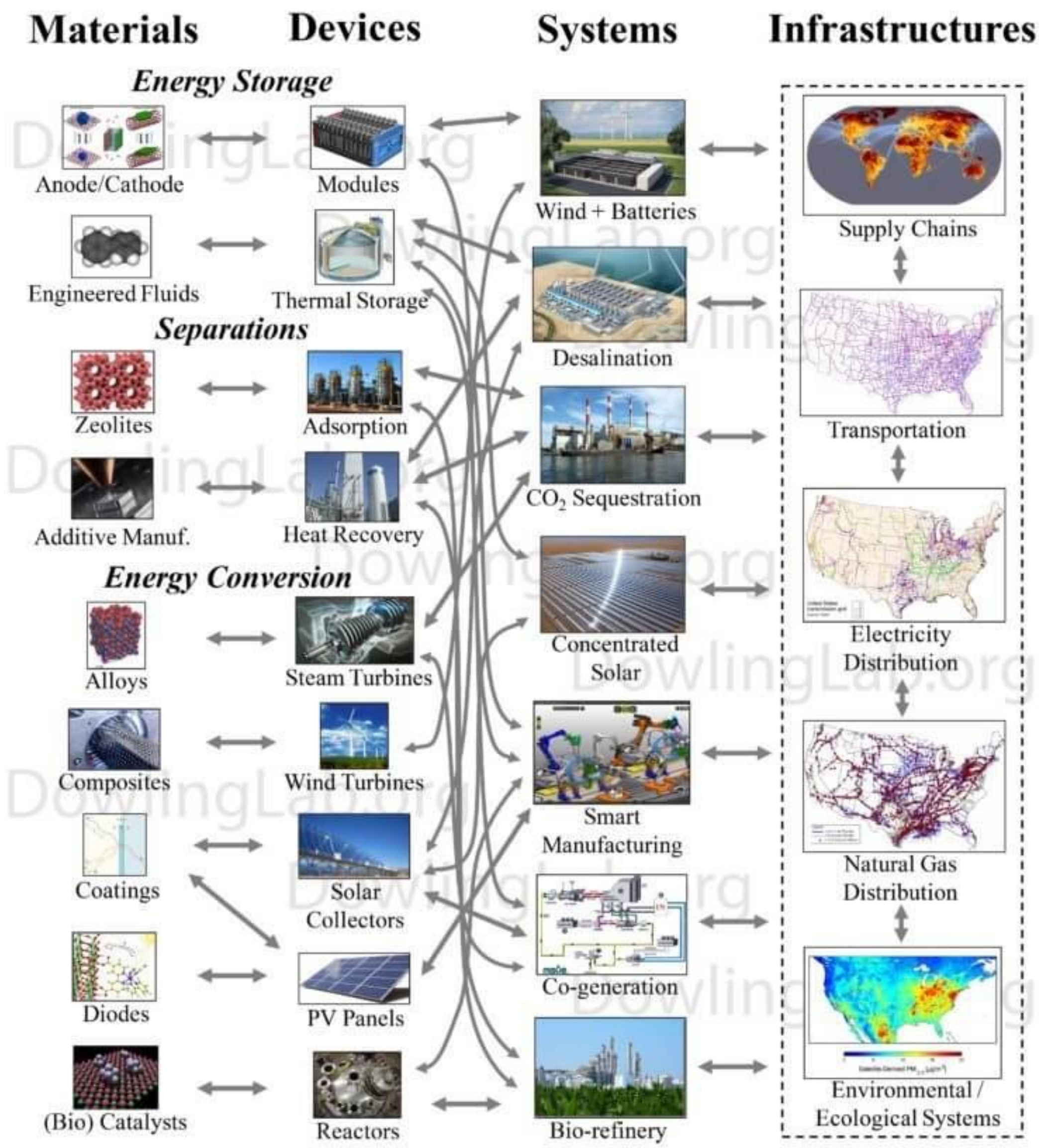
Department of Chemical and Biomolecular Engineering

University of Notre Dame

November 18, 2021

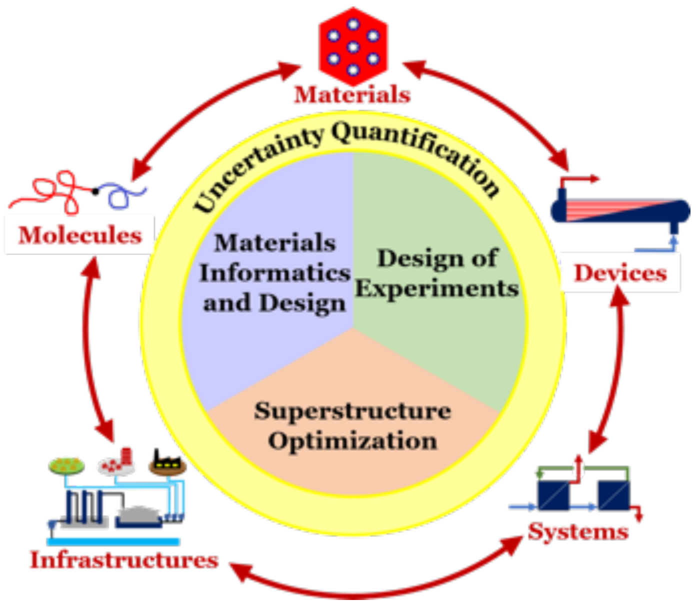
These slides are available at **http://psecommunity.org/LAPSE:2021.0803**

Research: Process Systems Engineering



Themes

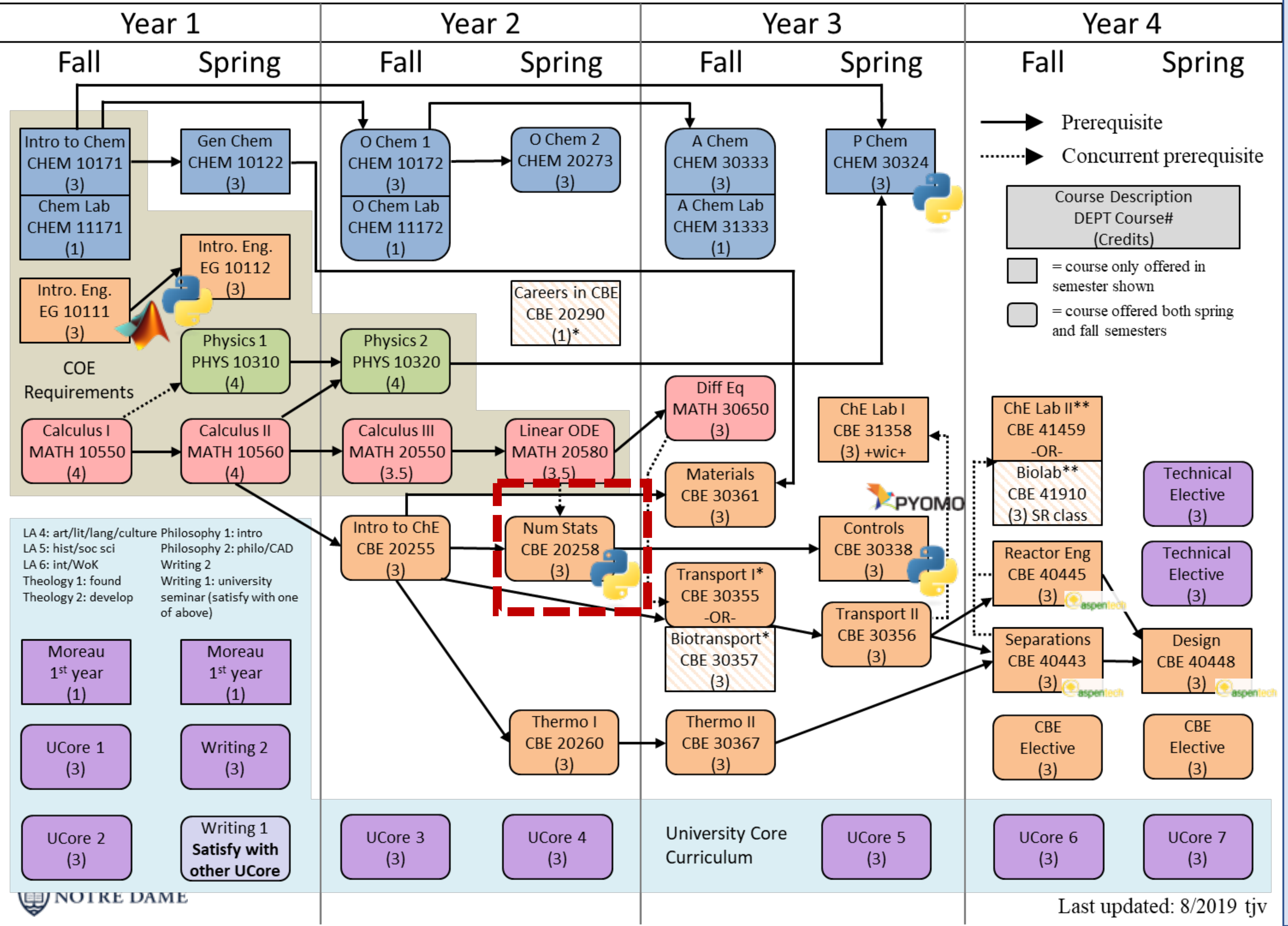
- Mathematical Modeling
- Computational Optimization
- Applied Statistics and Uncertainty Quantification
- Energy, Sustainability, & Systems Biology Applications


Materials
Uncertainty Quantification
Molecules
Devices
Materials Informatics and Design
Design of Experiments
Superstructure Optimization
Infrastructures
Systems

UNIVERSITY OF NOTRE DAME Funding:

Large Projects: IDAES CISTAR CCSI²

Chemical Engineering Suggested 4 Year Curriculum University of Notre Dame



Year 1 | Year 2 | Year 3 | Year 4

Fall | Spring | Fall | Spring | Fall | Spring | Fall | Spring

Intro to Chem
CHEM 10171
(3)

Chem Lab
CHEM 11171
(1)

Gen Chem
CHEM 10122
(3)

Intro. Eng.
EG 10111
(3)

Intro. Eng.
EG 10112
(3)

COE Requirements

Physics 1
PHYS 10310
(4)

Calculus I
MATH 10550
(4)

Calculus II
MATH 10560
(4)

O Chem 1
CHEM 10172
(3)

O Chem Lab
CHEM 11172
(1)

O Chem 2
CHEM 20273
(3)

Careers in CBE
CBE 20290
(1)*

Physics 2
PHYS 10320
(4)

Calculus III
MATH 20550
(3.5)

Linear ODE
MATH 20580
(3.5)

Intro to ChE
CBE 20255
(3)

Num Stats
CBE 20258
(3)

Thermo I
CBE 20260
(3)

A Chem
CHEM 30333
(3)

A Chem Lab
CHEM 31333
(1)

P Chem
CHEM 30324
(3)

Diff Eq
MATH 30650
(3)

Materials
CBE 30361
(3)

Transport I*
CBE 30355
-OR-
Biotransport*
CBE 30357
(3)

Thermo II
CBE 30367
(3)

ChE Lab I
CBE 31358
(3) +wic+

PYOMO

Controls
CBE 30338
(3)

Transport II
CBE 30356
(3)

ChE Lab II**
CBE 41459
-OR-
Biolab**
CBE 41910
(3) SR class

Reactor Eng
CBE 40445
(3)

Separations
CBE 40443
(3)

Technical Elective
(3)

Technical Elective
(3)

Design
CBE 40448
(3)

CBE Elective
(3)

CBE Elective
(3)

LA 4: art/lit/lang/culture
LA 5: hist/soc sci
LA 6: int/WoK
Theology 1: found
Theology 2: develop

Philosophy 1: intro
Philosophy 2: philo/CAD
Writing 2
Writing 1: university seminar (satisfy with one of above)

Moreau
1st year
(1)

Moreau
1st year
(1)

UCore 1
(3)

Writing 2
(3)

UCore 2
(3)

Writing 1
Satisfy with other UCore

UCore 3
(3)

UCore 4
(3)

University Core Curriculum

UCore 5
(3)

UCore 6
(3)

UCore 7
(3)

→ Prerequisite

⋯→ Concurrent prerequisite

Course Description
DEPT Course#
(Credits)

□ = course only offered in semester shown

▢ = course offered both spring and fall semesters

NOTRE DAME

Last updated: 8/2019 tjv

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

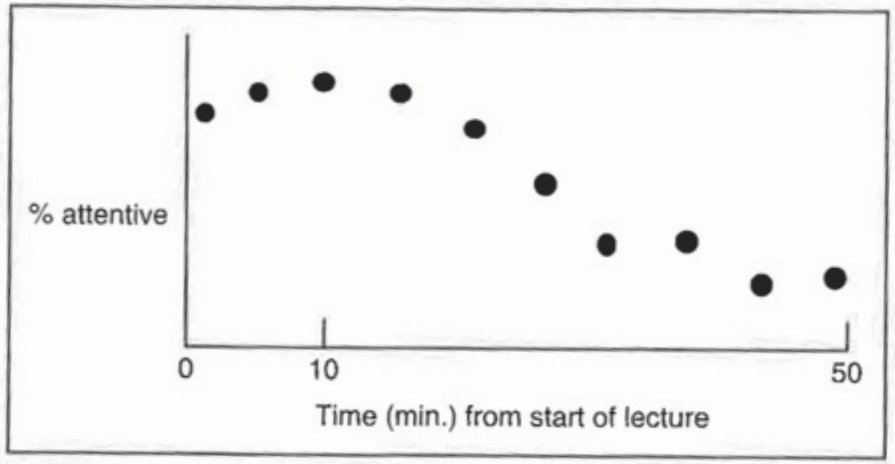


Figure 6.3–1: Attentiveness versus Time in Lecture—No Activities

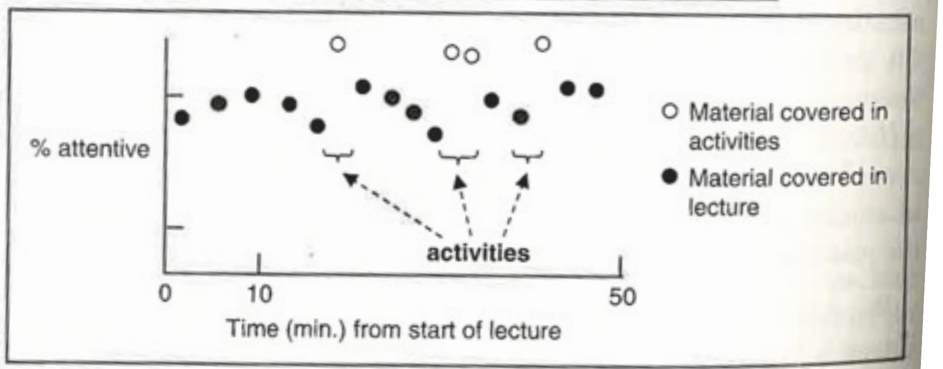


Figure 6.3–2: Attentiveness versus Time in Lecture—Activities Interspersed

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:

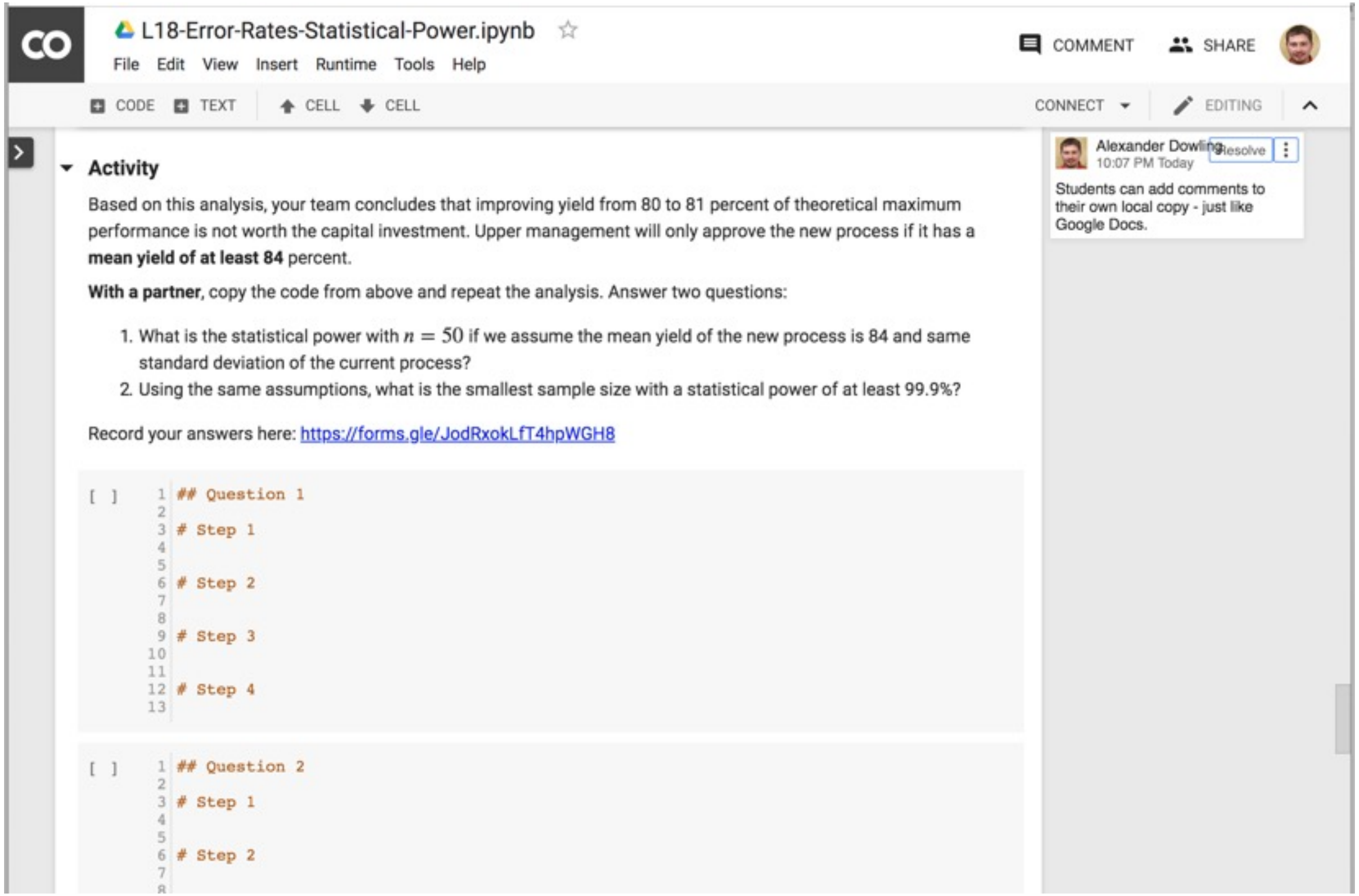
Like Google Docs, but for code

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


CO
L18-Error-Rates-Statistical-Power.ipynb
File Edit View Insert Runtime Tools Help
COMMENT
SHARE
CODE
TEXT
CELL
CELL
CONNECT
EDITING

Activity

Based on this analysis, your team concludes that improving yield from 80 to 81 percent of theoretical maximum performance is not worth the capital investment. Upper management will only approve the new process if it has a **mean yield of at least 84** percent.

With a partner, copy the code from above and repeat the analysis. Answer two questions:

1. What is the statistical power with $n = 50$ if we assume the mean yield of the new process is 84 and same standard deviation of the current process?
2. Using the same assumptions, what is the smallest sample size with a statistical power of at least 99.9%?

Record your answers here: https://forms.gle/JodRxokLfT4hpWGH8

```
[ ]
## Question 1

# Step 1


# Step 2


# Step 3


# Step 4

```

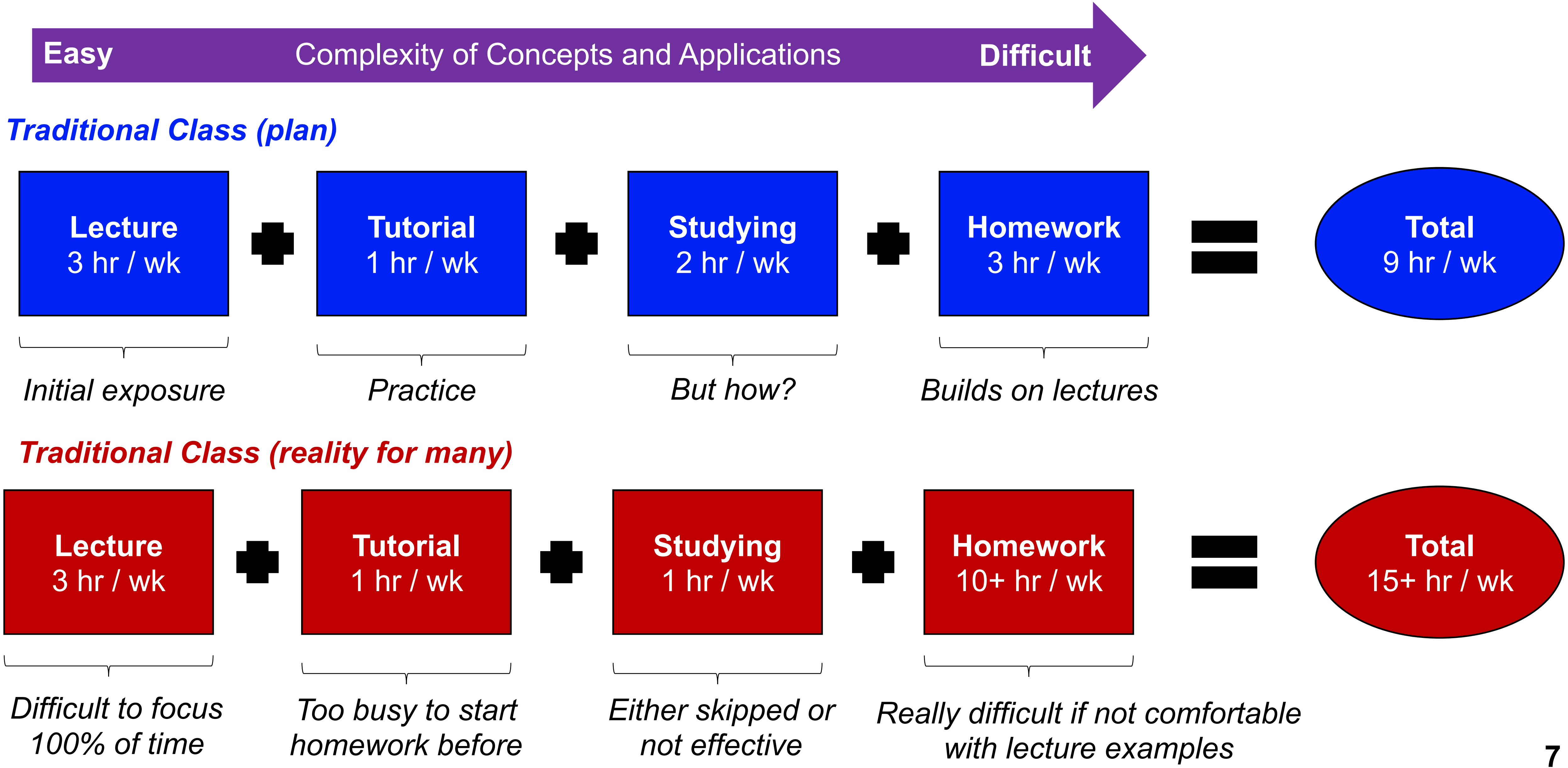
```
[ ]
## Question 2

# Step 1


# Step 2

```

Alexander Dowling
10:07 PM Today
Resolve

Students can add comments to their own local copy - just like Google Docs.

I show this slide on day 1.

Making your time more effective



Easy Complexity of Concepts and Applications **Difficult**

Traditional Class (plan)

Lecture 3 hr / wk + **Tutorial** 1 hr / wk + **Studying** 2 hr / wk + **Homework** 3 hr / wk = **Total** 9 hr / wk

Initial exposure

Practice

But how?

Builds on lectures

Traditional Class (reality for many)

Lecture 3 hr / wk + **Tutorial** 1 hr / wk + **Studying** 1 hr / wk + **Homework** 10+ hr / wk = **Total** 15+ hr / wk

Difficult to focus 100% of time

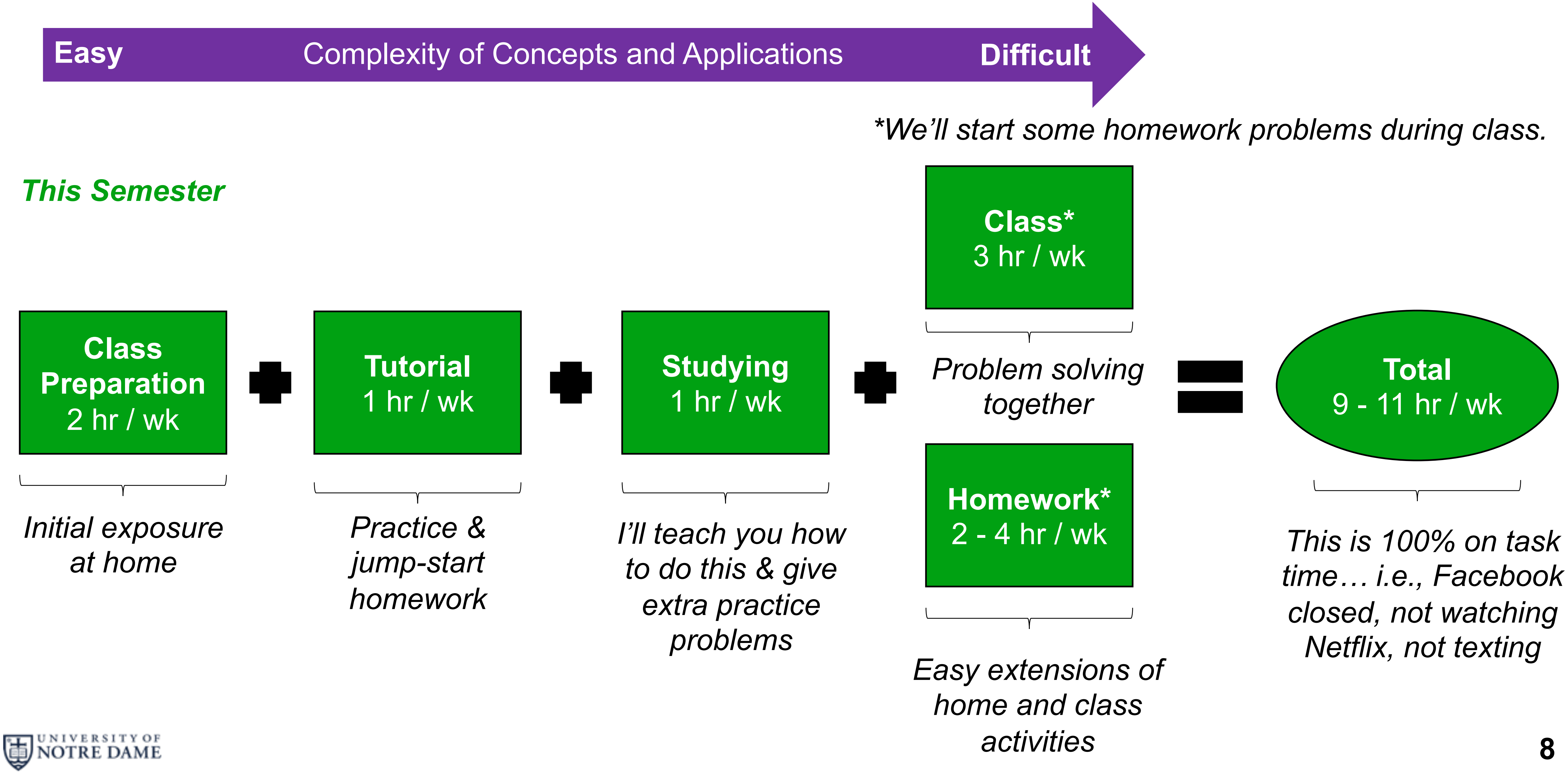
Too busy to start homework before

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:

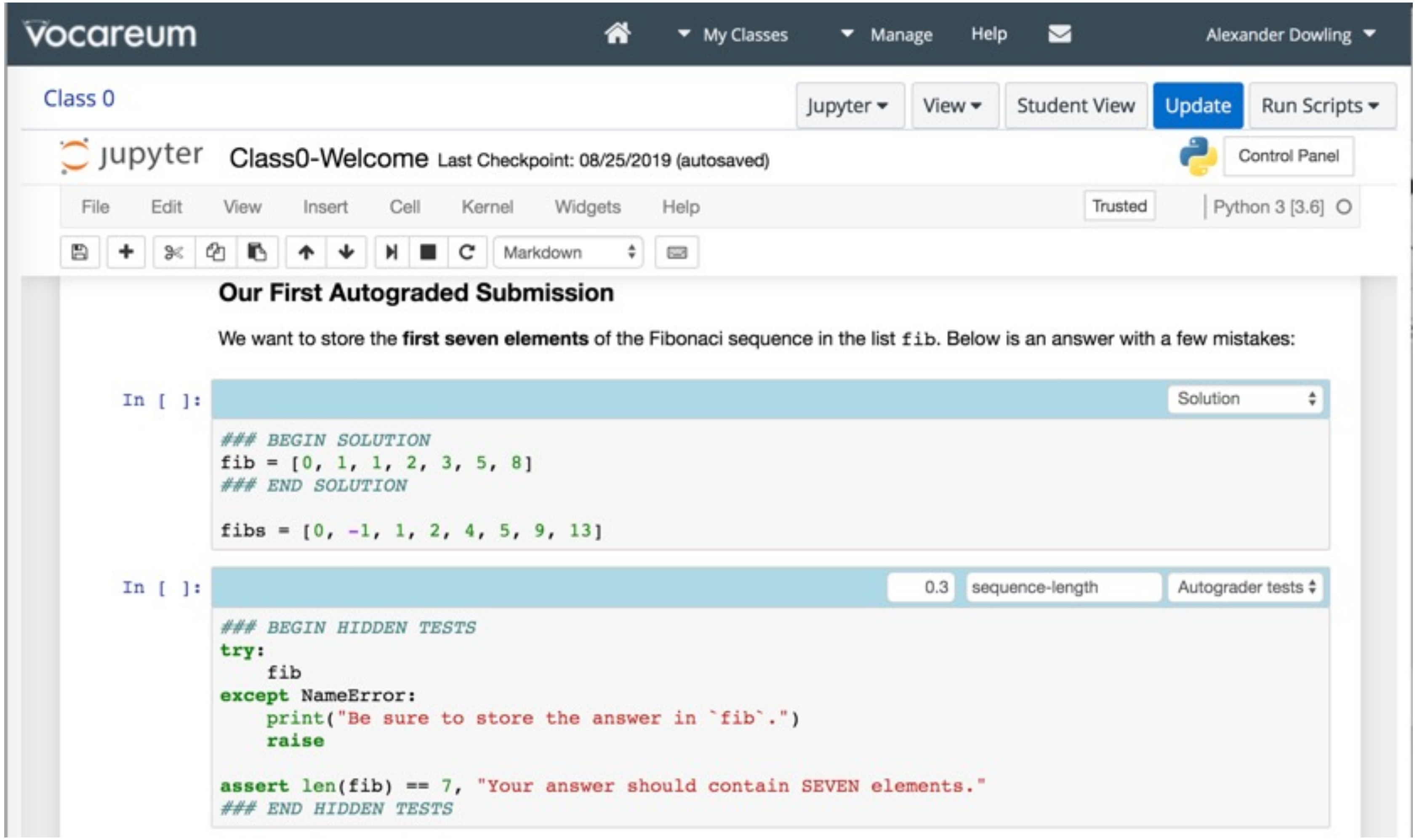
Many of the same cloud-based benefits as Colaboratory

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



Class0-Welcome Last Checkpoint: 08/25/2019 (autosaved)

Our First Autograded Submission

We want to store the **first seven elements** of the Fibonaci sequence in the list `fib`. Below is an answer with a few mistakes:

In [ ]: Solution

```
### BEGIN SOLUTION
fib = [0, 1, 1, 2, 3, 5, 8]
### END SOLUTION

fibs = [0, -1, 1, 2, 4, 5, 9, 13]
```

In [ ]: 0.3 sequence-length Autograder tests

```
### BEGIN HIDDEN TESTS
try:
    fib
except NameError:
    print("Be sure to store the answer in `fib`.")
    raise

assert len(fib) == 7, "Your answer should contain SEVEN elements."
### END HIDDEN TESTS
```

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

www.vocareum.com

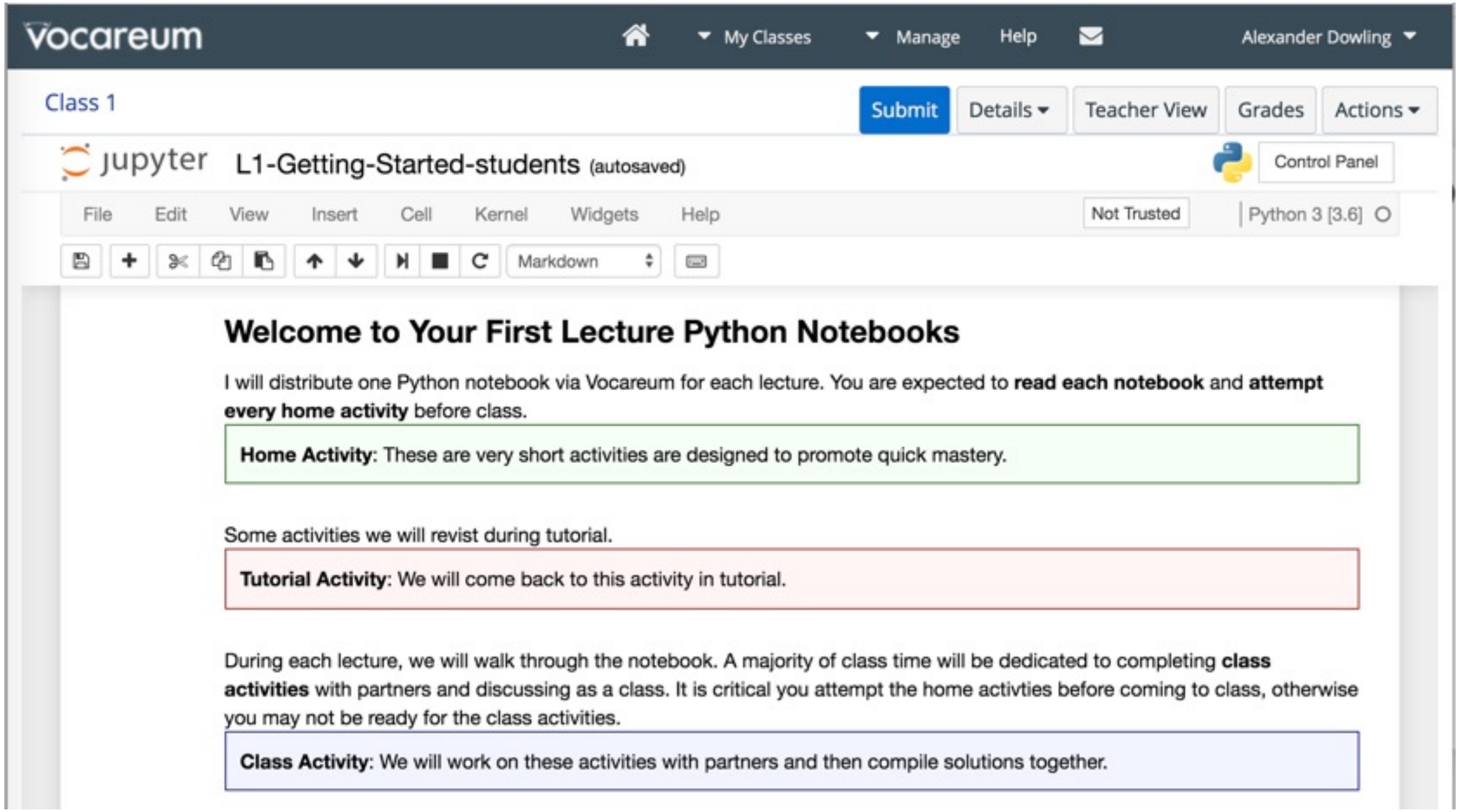
Benefits of Vocareum:

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)



Welcome to Your First Lecture Python Notebooks

I will distribute one Python notebook via Vocareum for each lecture. You are expected to **read each notebook** and **attempt every home activity** before class.

Home Activity: These are very short activities are designed to promote quick mastery.

Some activities we will revist during tutorial.

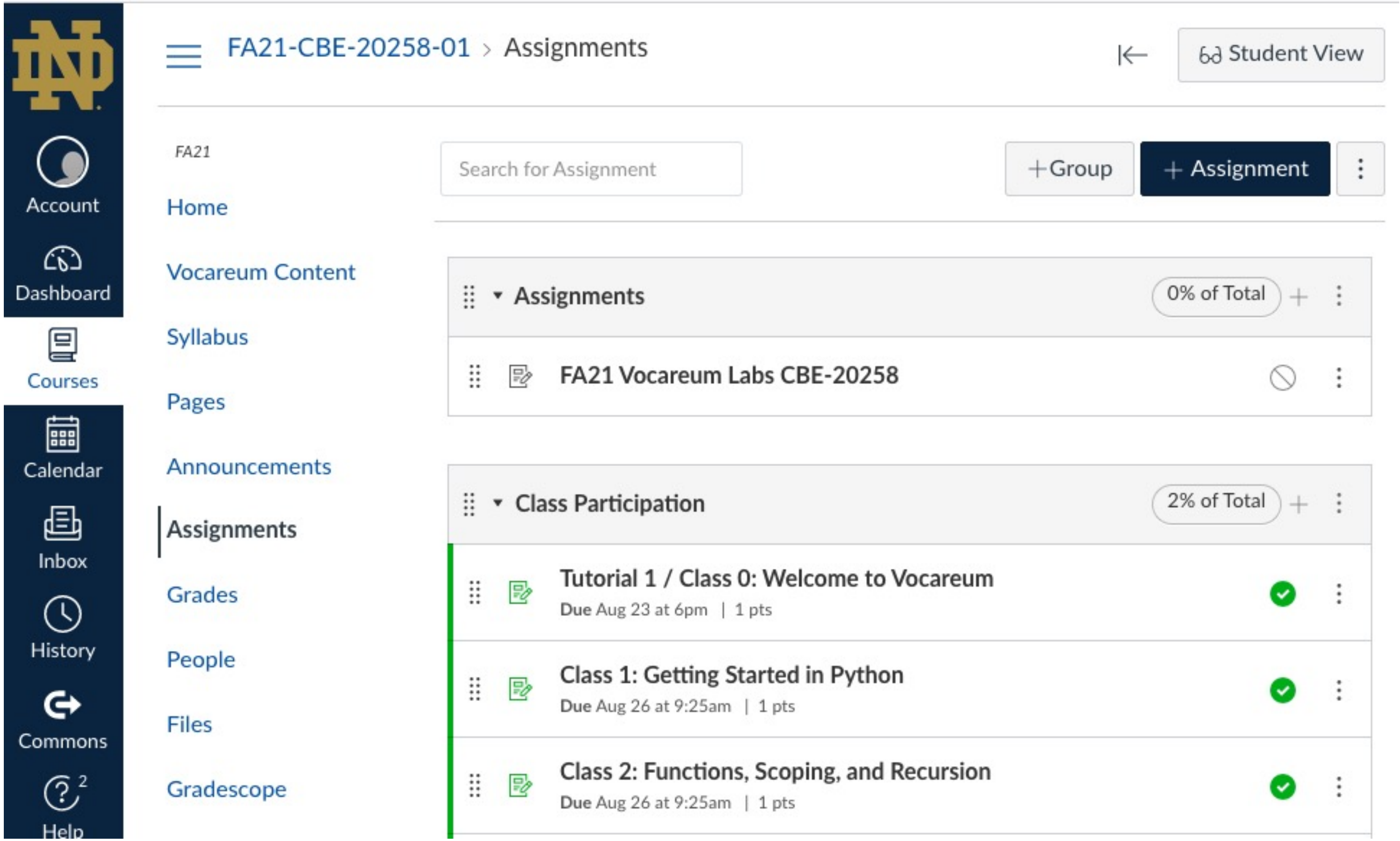
Tutorial Activity: We will come back to this activity in tutorial.

During each lecture, we will walk through the notebook. A majority of class time will be dedicated to completing **class activities** with partners and discussing as a class. It is critical you attempt the home activties before coming to class, otherwise you may not be ready for the class activities.

Class Activity: We will work on these activities with partners and then compile solutions together.

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



Assignments — 0% of Total

FA21 Vocareum Labs CBE-20258

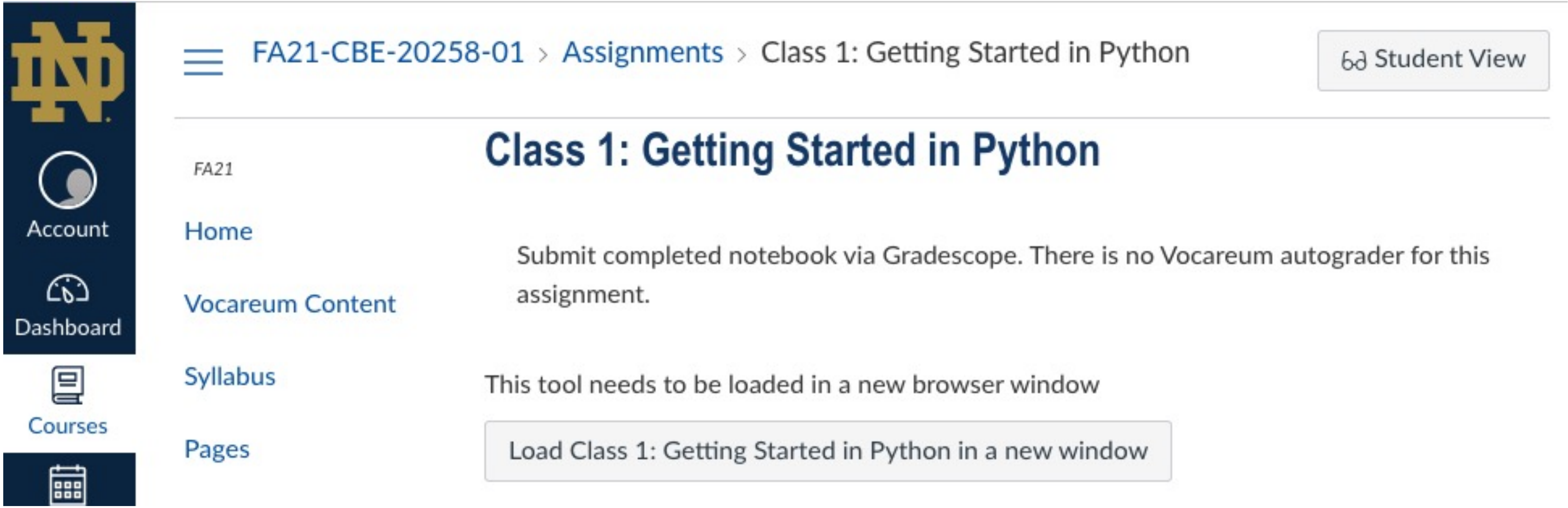
Class Participation — 2% of Total

Tutorial 1 / Class 0: Welcome to Vocareum
Due Aug 23 at 6pm | 1 pts

Class 1: Getting Started in Python
Due Aug 26 at 9:25am | 1 pts

Class 2: Functions, Scoping, and Recursion
Due Aug 26 at 9:25am | 1 pts

Each Canvas *Assignment* includes a link to Vocareum



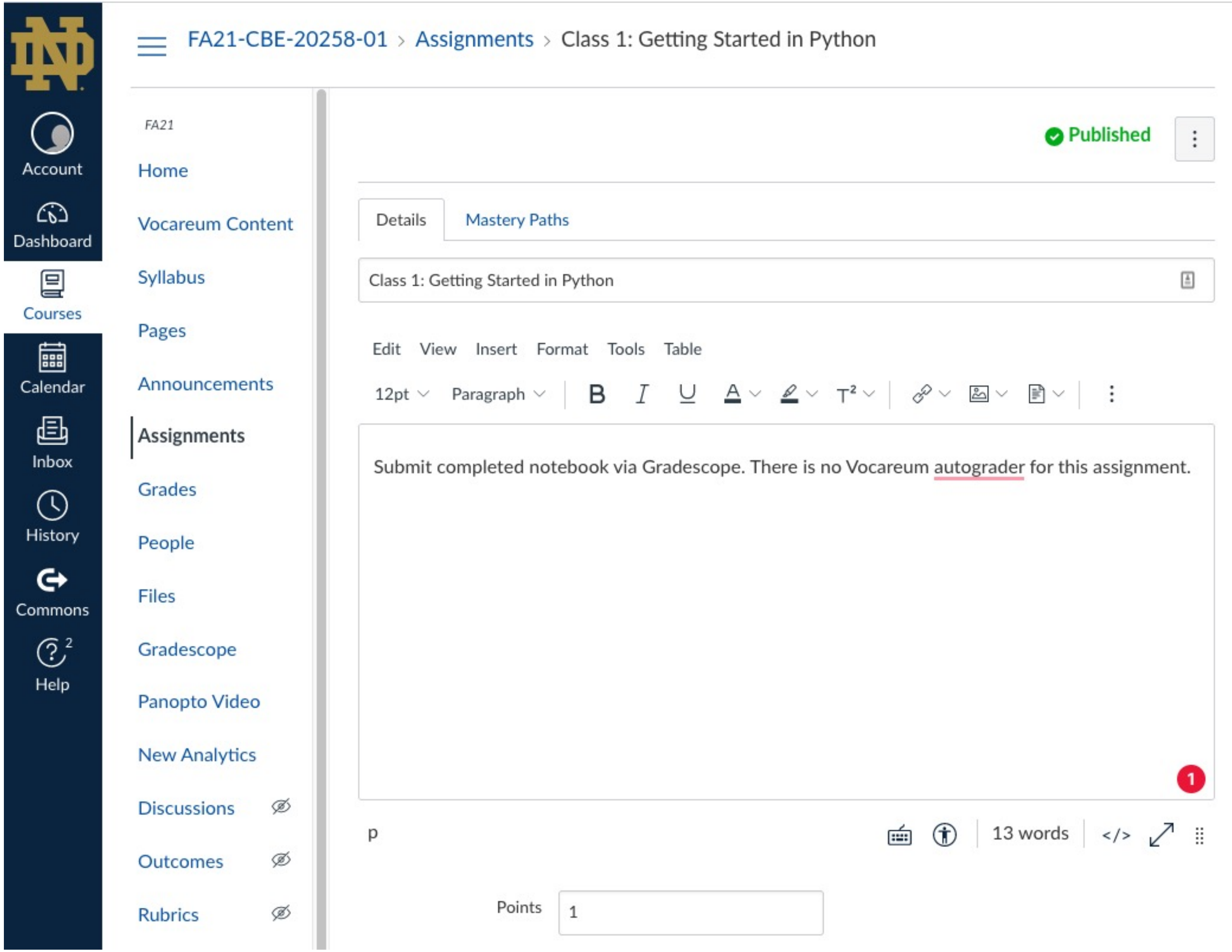
Class 1: Getting Started in Python

Submit completed notebook via Gradescope. There is no Vocareum autograder for this assignment.

This tool needs to be loaded in a new browser window

Load Class 1: Getting Started in Python in a new window

Instructor Manually Creates Each *Assignment* in Canvas



Class 1: Getting Started in Python

Submit completed notebook via Gradescope. There is no Vocareum autograder for this assignment.

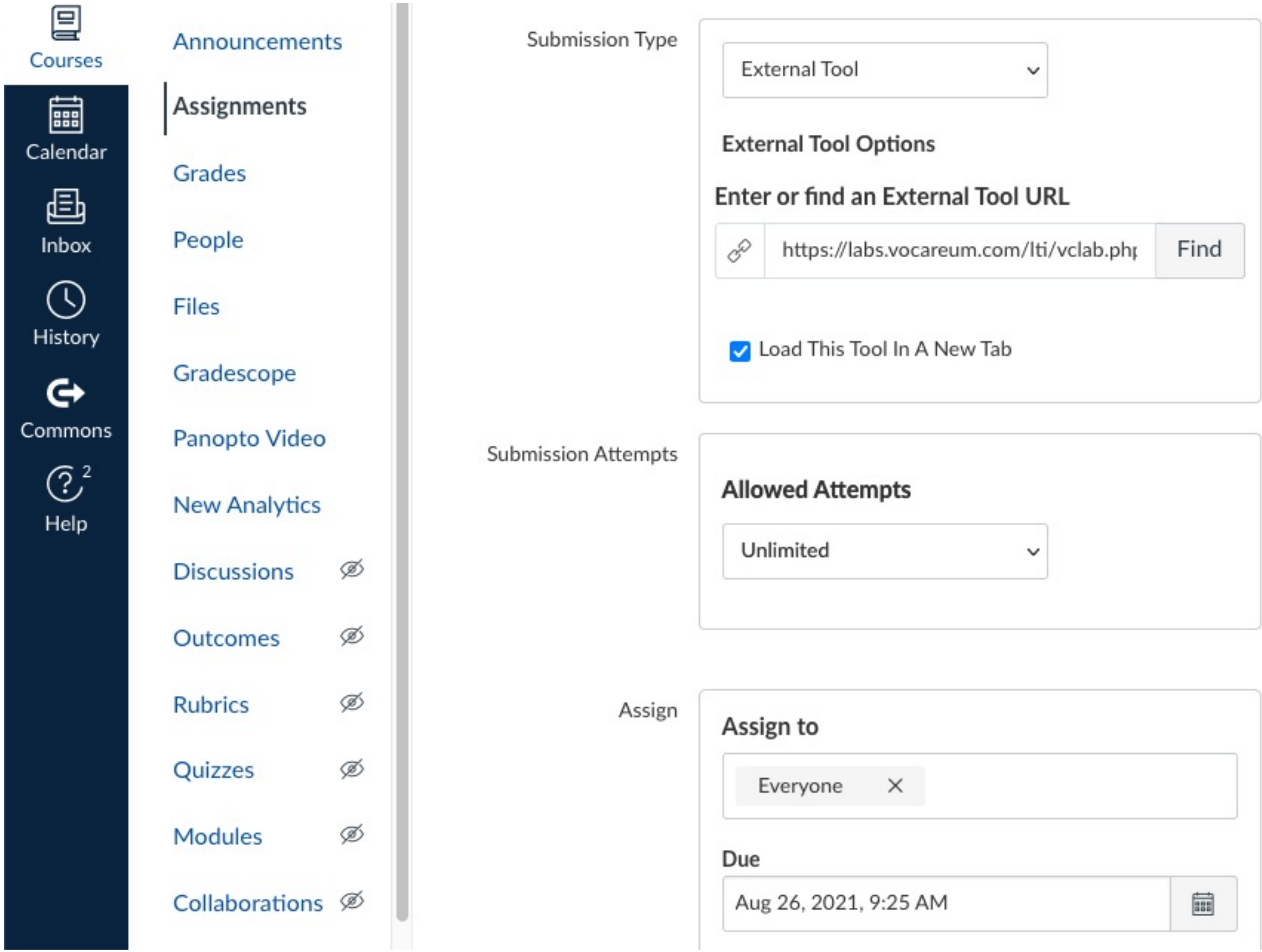
Points: 1

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



Submission Type

External Tool

External Tool Options

Enter or find an External Tool URL

https://labs.vocareum.com/lti/vclab.phן

☑ Load This Tool In A New Tab

Submission Attempts

Allowed Attempts

Unlimited

Assign

Assign to

Everyone ×

Due

Aug 26, 2021, 9:25 AM

Select "External Tool"

Copy this URL from Vocareum (easy)

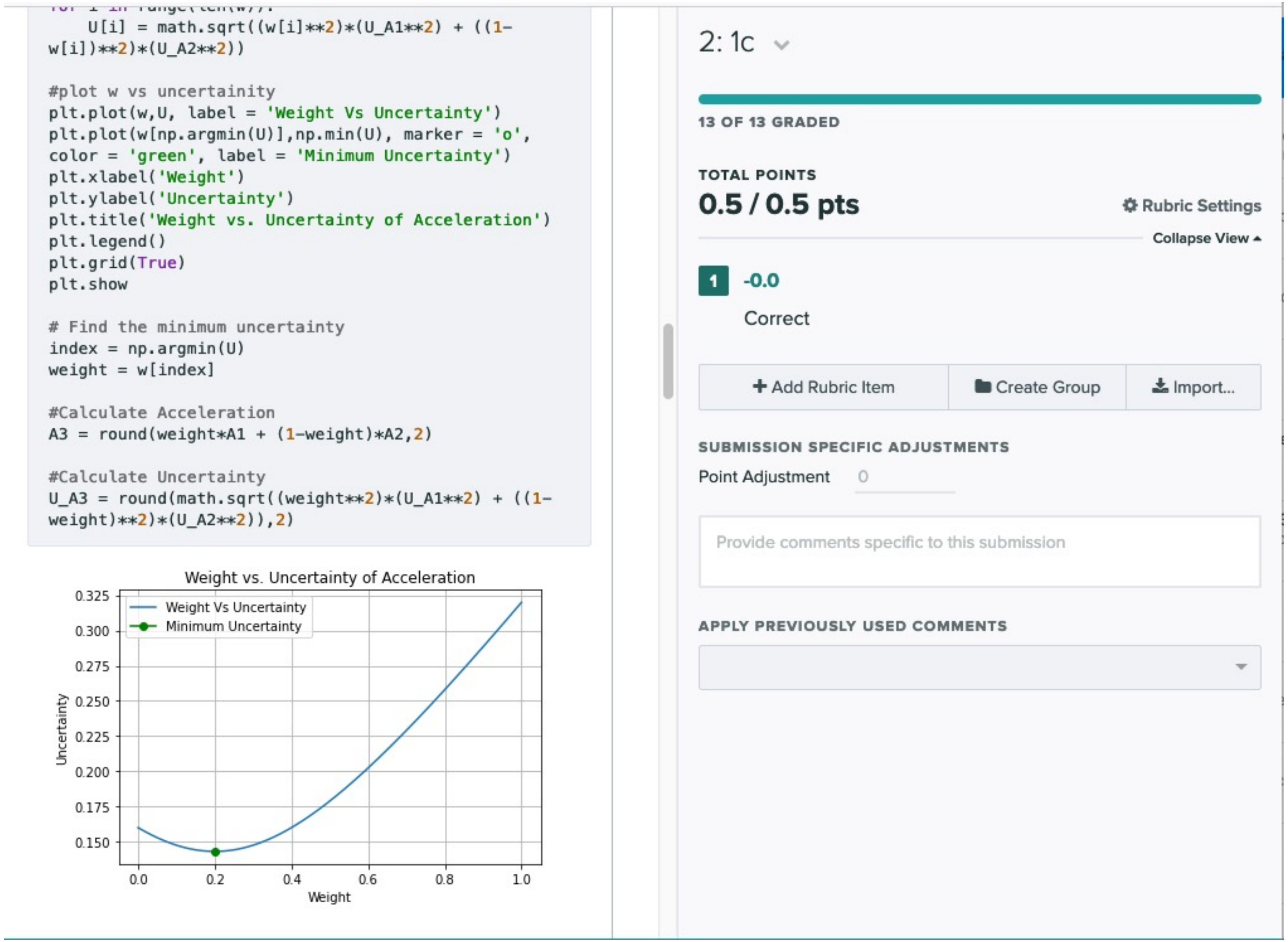
Check this box

Set due date here

Vocareum + Gradescope for Jupyter Notebooks

Typical assignments require three submissions:

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



```
for i in range(len(w)):
    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)
```

13 OF 13 GRADED

TOTAL POINTS

0.5 / 0.5 pts

1 -0.0

Correct

SUBMISSION SPECIFIC ADJUSTMENTS

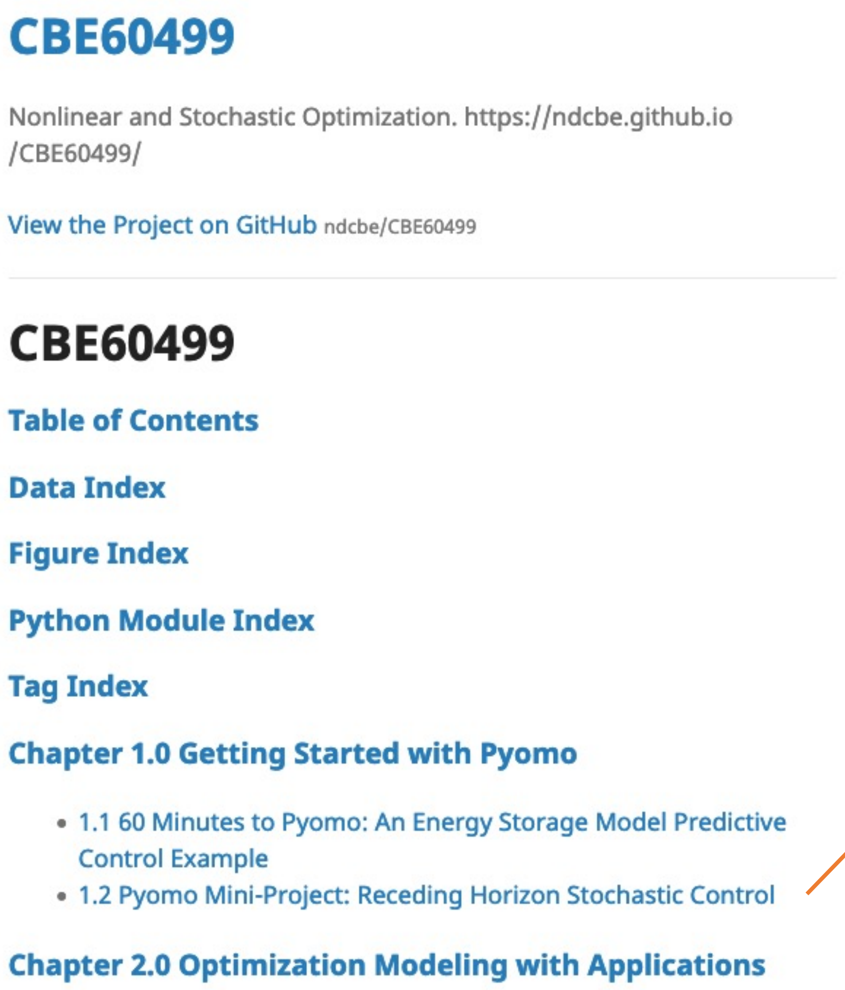
Point Adjustment 0

Provide comments specific to this submission

APPLY PREVIOUSLY USED COMMENTS

UNIVERSITY OF NOTRE DAME

nbpages + Google Colab



CBE60499

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

View the Project on GitHub ndcbe/CBE60499

CBE60499

Table of Contents

Data Index

Figure Index

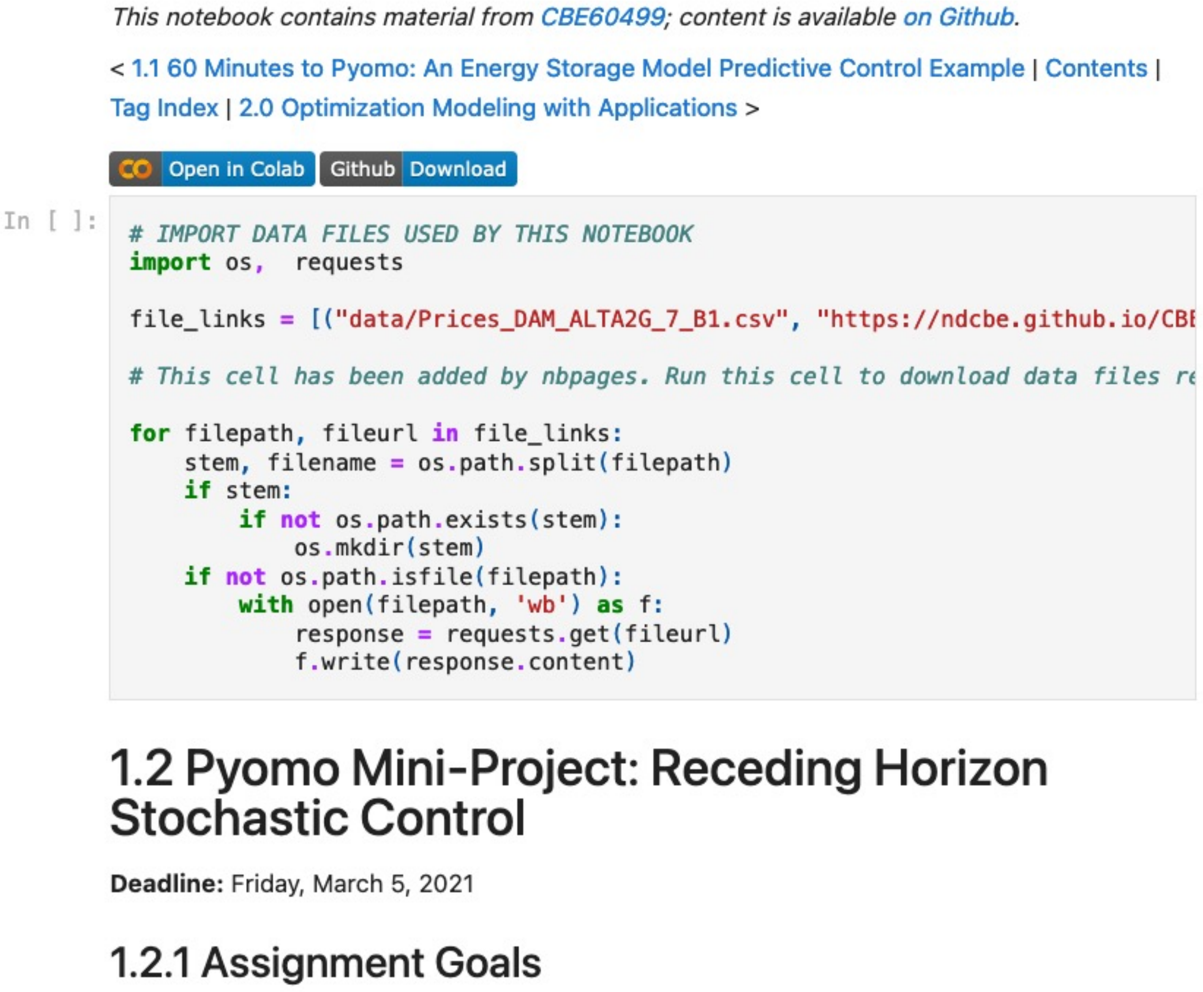
Python Module Index

Tag Index

Chapter 1.0 Getting Started with Pyomo

- 1.1 60 Minutes to Pyomo: An Energy Storage Model Predictive Control Example
- 1.2 Pyomo Mini-Project: Receding Horizon Stochastic Control

Chapter 2.0 Optimization Modeling with Applications



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 >

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

https://ndcbe.github.io/CBE60499/

https://github.com/jckantor/nbpages

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 text support

Great for computing focused undergraduate classes

colab

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

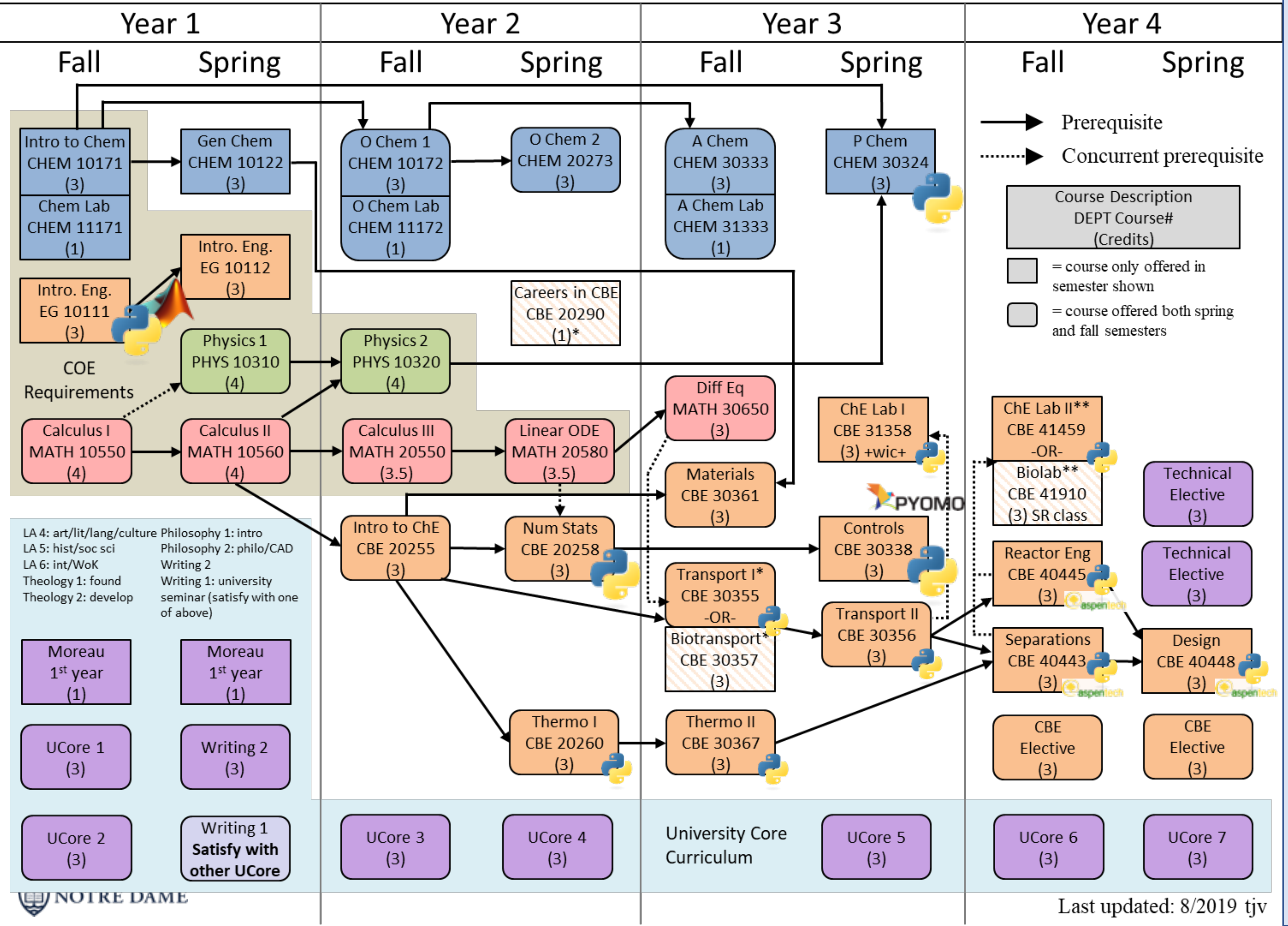
- Only manual grading via Gradescope

+ Easier to setup

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

Great for graduate classes and occasional class assignments/examples

Chemical Engineering Suggested 4 Year Curriculum University of Notre Dame



Year 1 — Fall | Spring

Year 2 — Fall | Spring

Year 3 — Fall | Spring

Year 4 — Fall | Spring

Intro to Chem CHEM 10171 (3)

Chem Lab CHEM 11171 (1)

Gen Chem CHEM 10122 (3)

O Chem 1 CHEM 10172 (3)

O Chem Lab CHEM 11172 (1)

O Chem 2 CHEM 20273 (3)

A Chem CHEM 30333 (3)

A Chem Lab CHEM 31333 (1)

P Chem CHEM 30324 (3)

Intro. Eng. EG 10111 (3)

Intro. Eng. EG 10112 (3)

Careers in CBE CBE 20290 (1)*

Physics 1 PHYS 10310 (4)

Physics 2 PHYS 10320 (4)

COE Requirements

Calculus I MATH 10550 (4)

Calculus II MATH 10560 (4)

Calculus III MATH 20550 (3.5)

Linear ODE MATH 20580 (3.5)

Diff Eq MATH 30650 (3)

Materials CBE 30361 (3)

ChE Lab I CBE 31358 (3) +wic+

PYOMO

ChE Lab II** CBE 41459 -OR- Biolab** CBE 41910 (3) SR class

Technical Elective (3)

Technical Elective (3)

Intro to ChE CBE 20255 (3)

Num Stats CBE 20258 (3)

Controls CBE 30338 (3)

Transport I* CBE 30355 -OR- Biotransport* CBE 30357 (3)

Transport II CBE 30356 (3)

Reactor Eng CBE 40445 (3)

Separations CBE 40443 (3)

Design CBE 40448 (3)

Thermo I CBE 20260 (3)

Thermo II CBE 30367 (3)

CBE Elective (3)

CBE Elective (3)

LA 4: art/lit/lang/culture
LA 5: hist/soc sci
LA 6: int/WoK
Theology 1: found
Theology 2: develop

Philosophy 1: intro
Philosophy 2: philo/CAD
Writing 2
Writing 1: university seminar (satisfy with one of above)

Moreau 1st year (1)

Moreau 1st year (1)

UCore 1 (3)

Writing 2 (3)

UCore 2 (3)

Writing 1 **Satisfy with other UCore**

UCore 3 (3)

UCore 4 (3)

University Core Curriculum

UCore 5 (3)

UCore 6 (3)

UCore 7 (3)

Last updated: 8/2019 tjv

Vision

Vertically integrate computing and statistics throughout the undergraduate curriculum

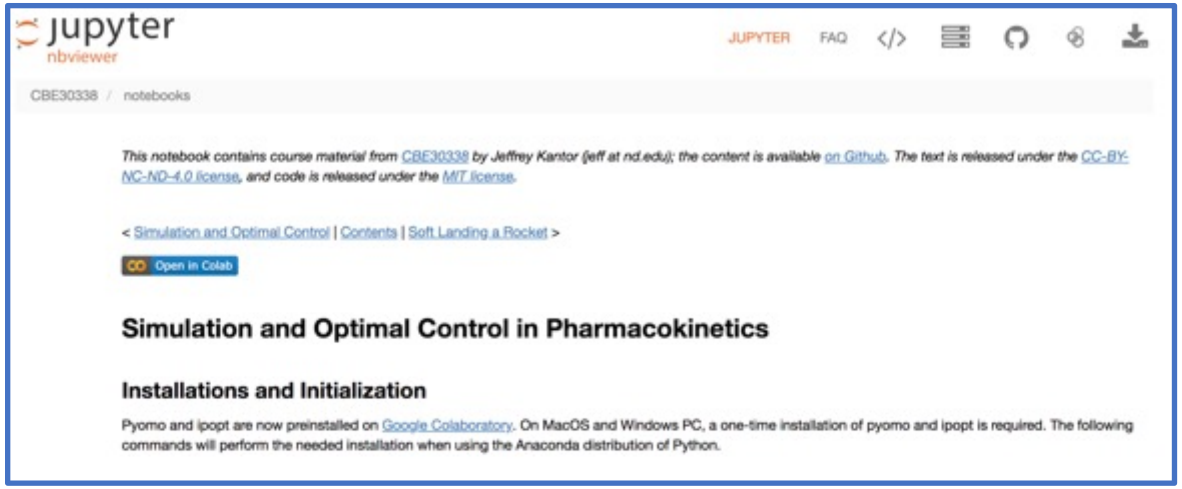
Library of Cloud-based Jupyter Notebooks

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

colab

vocareum

Special Thanks



CBE30338 / notebooks

This notebook contains course material from CBE30338 by Jeffrey Kantor (jeff at nd.edu); the content is available on Github. The text is released under the CC-BY-NC-ND-4.0 license, and code is released under the MIT license.

< Simulation and Optimal Control | Contents | Soft Landing a Rocket >

Open in Colab

Simulation and Optimal Control in Pharmacokinetics

Installations and Initialization

Pyomo and ipopt are now preinstalled on Google Colaboratory. On MacOS and Windows PC, a one-time installation of pyomo and ipopt is required. The following commands will perform the needed installation when using the Anaconda distribution of Python.

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 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`**