

Computing in Python

COMP 1800

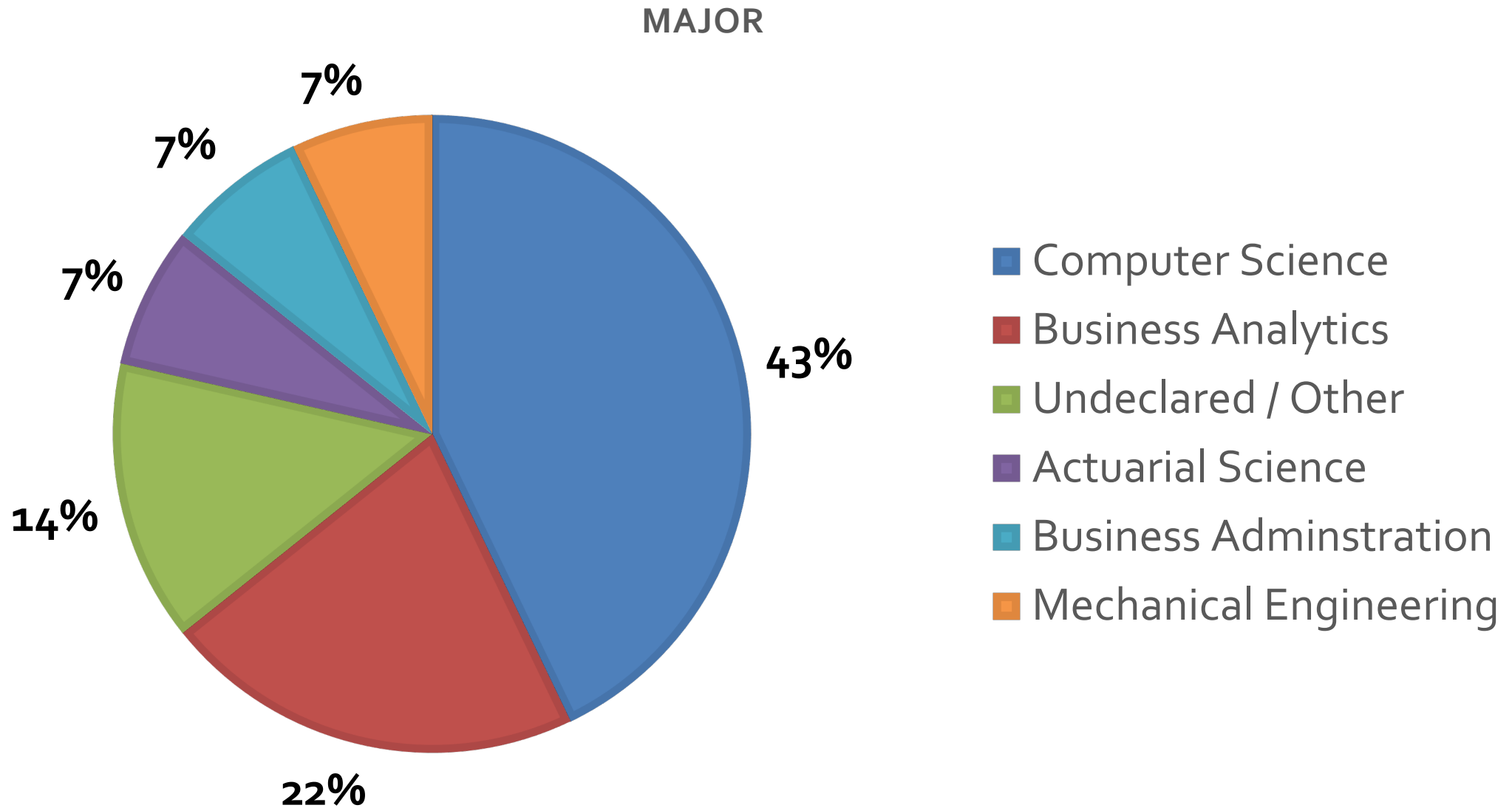
Who am I?

- Dr. Barry Wittman
- Not Dr. Barry Whitman
- Education:
 - PhD and MS in Computer Science, Purdue University
 - BS in Computer Science, Morehouse College
- Hobbies:
 - Reading, writing
 - Enjoying ethnic cuisine
 - DJing
 - Lockpicking
 - Stand-up comedy

How can you reach me?

- **E-mail:** `wittman1@otterbein.edu`
- **Office:** Point 105
- **Phone:** (614) 823-2944
- **Office hours:** **MWF** 1:45 – 3:00 p.m.,
WF 4:00 – 5:00 p.m.,
TR 1:00 – 4:00 p.m.,
and by appointment
- **Website:**
<http://faculty.otterbein.edu/wittman1/>

Who are you?



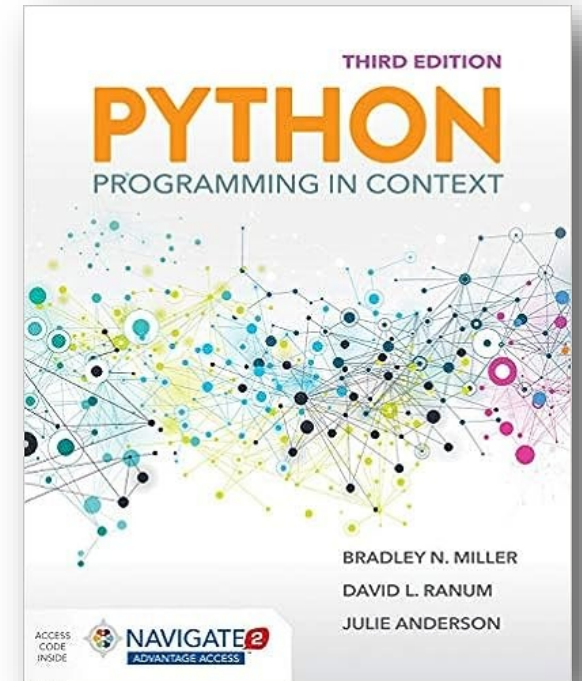
Why are we here?

- State your name, and give the salutation you would prefer:
Mr., Ms., or another
- What's the purpose of this class?
- What do you want to get out of it?
- Do you want to be here?
- Have you programmed before?

Course Overview

Textbook

- Bradley N. Miller, David L. Ranum, and Julie Anderson
- ***Python Programming in Context***
- 3rd Edition, 2019, Jones & Bartlett Learning
- ISBN-10: 1284175553
- ISBN-13: 978-1284175554



A note about the book...

- You absolutely need to read the book
- However, computer science is very much an applied science
- Reading the book is **not** enough
- You should be programming every day (or maybe every other day) to master the concepts

Course focus

- Problem solving
- Applications:
 - Image processing
 - Data processing
 - Simulation
 - Event-driven programming
- Features of Python:
 - Numbers
 - Functions
 - Strings
 - Collections
 - Files
 - Iteration
 - Regular expressions
 - Recursion
 - Classes and objects

More information

- For more information, visit the webpage:
<http://faculty.otterbein.edu/wittman1/comp1800>
- The webpage will contain:
 - The most current schedule
 - Notes available for download
 - Reminders about assignments and exams
 - Syllabus
 - Detailed policies and guidelines

Assignments

Ten assignments

- 50% of your grade will be ten assignments, programmed in Python
- Each assignment is individual
- Assignments are due every one to two weeks, depending on difficulty

Turning in assignments

- Assignments must be turned in by uploading them to Blackboard **before** the deadline
- Late assignments will not be accepted
 - Exception: Each student will have 3 grace days
 - You can use the grace days together or separately as extensions for your assignments
 - You must inform me **before** the deadline that you are going to use grace days

Quizzes

Pop Quizzes

- 5% of your grade will be pop quizzes
- These quizzes will be based on material covered in the previous one or two lectures
- They will be graded leniently
- They are useful for these reasons:
 1. Informing me of your understanding
 2. Feedback to you about your understanding
 3. Easy points for you
 4. Attendance

Exams

Exams

- There will be two equally weighted exams totaling 30% of your final grade
 - Exam 1: 09/25/2023
 - Exam 2: 10/30/2023
- The final exam will be worth 15% of your grade
 - Final: 2:45 – 4:45 p.m.
12/08/2023
- All exams will be in our normal classroom

Exam format

- Conceptual portion
 - Multiple choice and short answer
- Programming portion
 - Short programming problems you will write code for

Course Schedule

Tentative schedule

Week	Starting	Topics	Chapters	Notes
1	08/21/23	Introduction	1	
2	08/28/23	Numbers and functions	2	
3	09/04/23	Computation		
4	09/11/23	Characters and strings	3	
5	09/18/23	Collections	4	
6	09/25/23	Files and iteration	5	Exam 1
7	10/02/23	Image processing	6	
8	10/09/23	More image processing	7	October Break
9	10/16/23	Data processing	8	
10	10/23/23	Regular expressions		
11	10/30/23	Recursion	9	Exam 2
12	11/06/23	Classes and objects	10	
13	11/13/23	Simulation	11	
14	11/20/23	Event-driven Programming	13	Thanksgiving
15	11/27/23	Review	All	

Policies

Grading breakdown

50%

- Ten assignments

5%

- Quizzes

30%

- Two equally weighted midterm exams

15%

- Final exam

Grading scale

A	93-100	B-	80-82	D+	67-69
A-	90-92	C+	77-79	D	60-66
B+	87-89	C	73-76	F	60-62
B	83-86	C-	70-72		

Attendance

- You are expected to attend class
- You are expected to have read the material we are going to cover **before** class
- Missed quizzes cannot be made up
- Exams must be made up **before** the scheduled time, for excused absences

R-E-S-P-E-C-T

- I hate having a slide like this
- I ask for respect for your classmates and for me
- You are smart enough to figure out what that means
- A few specific points:
 - Silence communication devices
 - **Don't use the computers except when explicitly asked to**
 - No food or drink in the lab

Computer usage

- Sometimes, I will do code examples in class
- In those cases, you are welcome to follow along
- However, students are always tempted to browse the Internet, etc.
- Research shows that it is nearly impossible to do two things at the same time (e.g. use Reddit and listen to a lecture)
- For your own good, I will enforce this by taking **1% of your final grade** every time I catch you using your computer for anything other than coding

Academic dishonesty

- Don't cheat
- **First offense:**
 - I will try to give you a zero for the assignment, then lower your final letter grade for the course by one full grade
- **Second offense:**
 - I will try to fail you for the course and kick you out of Otterbein University
- Refer to the Student Handbook for the official policy
- Ask me if you have questions or concerns
- **You are never allowed to look at another student's code**
- **Don't use AI tools like ChatGPT to write any code that you turn in**
- **I will use tools that automatically test code for similarity**

Programming assignments

- Must be handed in on time
 - If your program is late (and grace days are not available), it will score zero points
- Must be done individually
 - If I can ascertain that code from one student's assignment appears in another student's assignment, both assignments will score zero points
 - Both students will also have a full letter grade reduction at the end of the semester

Disability Services

- The University has a continuing commitment to providing access and reasonable accommodations for students with disabilities, including mental health diagnoses and chronic or temporary medical conditions. Students who may need accommodations or would like referrals to explore a potential diagnosis are urged to contact Disability Services (DS) as soon as possible. DS will facilitate accommodations and assist the instructor in minimizing barriers to provide an accessible educational experience. Please contact DS at DisabilityServices@otterbein.edu. More info can also be found [here](#). Your instructor is happy to discuss accommodations privately with you as well.

A Taste of the Course

What is this course about?

- COMP 1600 teaches programming in Java
 - But it's really trying to teach *programming*
- This course is more about *using* Python
- Sure, you'll learn a bit about programming
 - But programming is a big subject
- We'll give you enough skills to be dangerous

Things to remember about computers

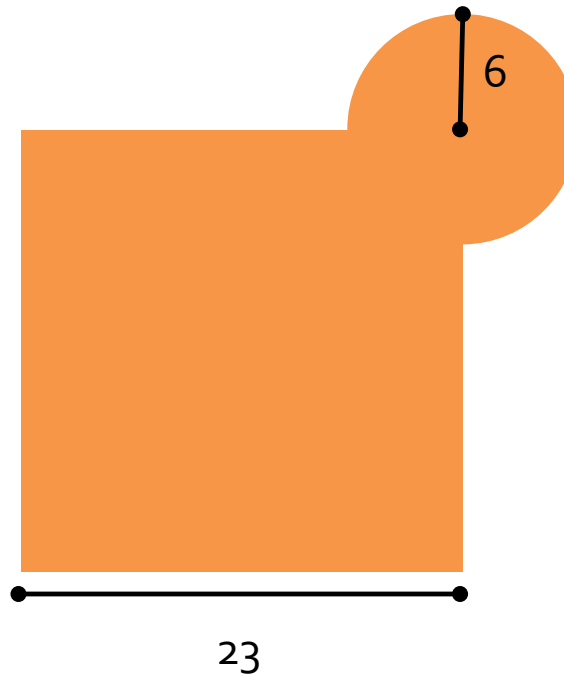
- The book wants you to keep in mind six things about computers:
 1. Computers are dumb
 2. Computers only do what you tell them to do
 3. Computers do what you tell them to do really fast
 4. Computers don't remember anything unless you tell them how to remember
 5. Computers do what you tell them to do, not what you want them to do
 6. Computers only do what they're told, in exactly the order they're told

Problem solving

- The famous mathematics educator George Pólya outlined a series of steps for solving problems:
 1. Understand the problem
 2. Make a plan
 3. Execute the plan
 4. Look back and reflect

Applying Pólya's problem solving method

- Use the Pólya's method to find the total area and perimeter of the shape below, which is made up of a circle and a square:



Solving problems with computers

- One of the things that makes solving problems with computers challenging is that we usually don't want an answer to **one problem**
- We want a program that will solve **many** similar problems
- For example, we could write a program that takes in the size of *any* square and *any* circle arranged like before and finds the correct area and perimeter

Algorithms

- What's an algorithm?
- A finite set of steps you can follow to solve a problem
- In terms of the previous slide, it's a finite answer to an infinite number of problems
- Can you give an example?
- Long division

Software engineering

- One version of the waterfall model of software engineering follows Pólya's problem-solving steps:
 1. Understand the problem
 2. Write an algorithm (Make a plan)
 3. Turn the algorithm into code (Execute the plan)
 4. Test and debug (Look back and reflect)

Problem-solving strategies

- The book gives three problem-solving strategies that can be useful for finding general solutions to all kinds of problems:
 - **Simplification:** Take a complicated example and turn it into a simpler, smaller example when checking a solution
 - **Generalization:** Sort of the opposite, use lots of examples and try to find a pattern
 - **Representation:** Find a good way to visualize the problem, maybe with diagrams, graphs, or mathematical notation

A Brief Taste of IDLE

Upcoming

Next time...

- Overview of Python

Reminders

- Read Chapter 1
- I recommend installing Python on your computer
 - Available for free here: <https://www.python.org/downloads/>
 - You should always be able to use the lab, but it's good to have a backup option
 - Come to office hours if you need assistance!