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:** wittmanb@etown.edu
- **Office:** Esbenshade 284B
- **Phone:** (717) 361-4761
- **Office hours:** MWF 11:00am – 12:00pm  
  MWF 2:00 – 3:20pm  
  T 1:00 – 3:00pm  
  And by appointment
- **Website:**  
  http://users.etown.edu/w/wittmanb/
Who are you?

Major

- Freshman/Undeclared
- Actuarial Science
- Business
- Mathematics Education
- Accounting
- Engineering
- Mathematical Business

- 57%
- 19%
- 5%
- 5%
- 5%
- 5%
- 5%
Why are we here?

- 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
Barry Wittman, Aditya Mathur, and Tim Korb

*Start Concurrent: An Introduction to Problem Solving in Java with a Focus on Concurrency*

- ISBN-10: 1626710090
Yes, I feel guilty about making you buy my book
  - But it's not very expensive...
The book's not bad
  - Your feedback is highly valued for the next edition
I highly encourage you to read it
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
Computer science

- What’s a computer?
- What’s computer science?
- What’s a computer program?
What’s an algorithm?
A finite set of steps you can follow to solve a problem
Can you give an example?
Long division
What’s a programming language?

Examples:
- C++
- Java
- PHP
- ML
- Prolog

A language we can use to encode an algorithm into a form a computer can execute
What’s a compiler?

Turns the (human readable) programming language into the (computer readable) instructions that a computer executes

For Java, we'll be using Eclipse as our compiler
Course focus

- Problem solving (writing the algorithm)
- Converting the algorithm to Java
- Features of Java we will focus on:
  - Variables
  - Mathematical operations
  - Selection statements (making choices)
  - Loops (repeating things)
  - Methods (dividing work into chunks)
  - Arrays (lists of things)
  - Input and output
  - Classes and object orientation
  - Searching and sorting
  - Image and audio processing
For more information, visit the webpage:
http://users.etown.edu/w/wittmanb/cs121

The webpage will contain:
- The most current schedule
- Notes available for download
- Reminders about projects and exams
- Syllabus (you can request a printed copy if you like)
- Detailed policies and guidelines

Piazza will allow for discussion and questions about the projects:
https://piazza.com/etown/fall2017/cs121
Projects
35% of your grade will be five projects. Each will focus on a different area from the course:

- I/O and arithmetic
- `if` statements
- Loops
- Methods and arrays
- Sorting, searching, and image processing

Each project is individual.
Turning in projects

- Projects must be turned in by uploading them to Canvas ([https://etown.instructure.com/](https://etown.instructure.com/)) before the deadline
- Do not put projects in your public directories
- Late projects 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 projects
  - You must inform me before the deadline that you are going to use grace days
- Assignments that don't compile get 0 points
Labs
In-class Programming Exercises
Labs

- 15% of your grade will be based around programming labs
- Labs are on Friday
  - Except for the weeks of Thanksgiving and October Break, when they're on Wednesday
- Approximately one hour of class will be devoted to each lab
- Each lab will focus on the solution of a problem
- Work should be done individually, but the goal is to learn, and I will help everyone
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 in-class exams totaling 30% of your final grade
  - Exam 1: 09/25/2017
  - Exam 2: 11/06/2017
- The final exam will be worth 15% of your grade
  - Final: 2:30 – 5:30pm
    12/11/2017
Exam format

- Conceptual portion
  - Multiple choice and short answer
- Programming portion
  - Short programming problems you will write code for
<table>
<thead>
<tr>
<th>Week</th>
<th>Starting</th>
<th>Topics</th>
<th>Chapters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>08/28/17</td>
<td>Introduction</td>
<td>1 and notes</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>09/04/17</td>
<td>Primitive data types</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>09/11/17</td>
<td>Basic operations</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>09/18/17</td>
<td>Conditionals</td>
<td>4</td>
<td>Project 1 Due</td>
</tr>
<tr>
<td>5</td>
<td>09/25/17</td>
<td>Loops</td>
<td>5</td>
<td>Exam 1</td>
</tr>
<tr>
<td>6</td>
<td>10/02/17</td>
<td>More loops</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>10/09/17</td>
<td>Arrays</td>
<td>6</td>
<td>Project 2 Due</td>
</tr>
<tr>
<td>8</td>
<td>10/16/17</td>
<td>Static methods</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10/23/17</td>
<td>Advanced method use</td>
<td>8</td>
<td>Project 3 Due</td>
</tr>
<tr>
<td>10</td>
<td>10/30/17</td>
<td>Classes and objects</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11/06/17</td>
<td>Custom classes</td>
<td>9</td>
<td>Exam 2</td>
</tr>
<tr>
<td>12</td>
<td>11/13/17</td>
<td>Searching and sorting</td>
<td>Notes</td>
<td>Project 4 Due</td>
</tr>
<tr>
<td>13</td>
<td>11/20/17</td>
<td>Image processing</td>
<td>Notes</td>
<td>Thanksgiving</td>
</tr>
<tr>
<td>14</td>
<td>11/27/17</td>
<td>Advanced topics</td>
<td>11 and 20</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12/04/17</td>
<td>Review</td>
<td>All</td>
<td>Project 5 Due</td>
</tr>
</tbody>
</table>
Project schedule

- **Project 1:** 5%  Tentatively due **9/22/2017**
- **Project 2:** 6%  Tentatively due **10/09/2017** (Monday)
- **Project 3:** 7%  Tentatively due **10/27/2017**
- **Project 4:** 7%  Tentatively due **11/17/2017**
- **Project 5:** 10% Tentatively due **12/08/2017**
Policies
Grading breakdown

- 35% • Five projects
- 15% • Labs (in-class programming)
- 5% • Quizzes
- 30% • Two equally weighted midterm exams
- 15% • Final exam
# Grading scale

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
<th>Grade</th>
<th>Range</th>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93-100</td>
<td>B-</td>
<td>80-82</td>
<td>D+</td>
<td>67-69</td>
</tr>
<tr>
<td>A-</td>
<td>90-92</td>
<td>C+</td>
<td>77-79</td>
<td>D</td>
<td>63-66</td>
</tr>
<tr>
<td>B+</td>
<td>87-89</td>
<td>C</td>
<td>73-76</td>
<td>D-</td>
<td>60-62</td>
</tr>
<tr>
<td>B</td>
<td>83-86</td>
<td>C-</td>
<td>70-72</td>
<td>F</td>
<td>0-59</td>
</tr>
</tbody>
</table>
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 and labs must be made up before the scheduled time, for excused absences
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
Sometimes, I will do code examples in class. In those cases, you are welcome to follow along. However, students are always tempted to surf the Internet, etc. Research shows that it is nearly impossible to do two things at the same time (e.g. use Facebook 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 give you a zero for the assignment, then lower your final letter grade for the course by one full grade
- **Second offense:**
  - I will fail you for the course and try to kick you out of Elizabethtown College
- 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**
- **I will use tools that automatically test code for similarity**
**Programming projects**

- **Must compile**
  - If your program does not compile, it will score zero points
- **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 project appears in another student’s project, both projects will score zero points
  - Both students will also have a full letter grade reduction at the end of the semester
Elizabethtown College welcomes otherwise qualified students with disabilities to participate in all of its courses, programs, services, and activities. If you have a documented disability and would like to request accommodations in order to access course material, activities, or requirements, please contact the Director of Disability Services, Lynne Davies, by phone (361-1227) or e-mail daviesl@etown.edu. If your documentation meets the college’s documentation guidelines, you will be given a letter from Disability Services for each of your professors. Students experiencing certain documented temporary conditions, such as post-concussive symptoms, may also qualify for temporary academic accommodations and adjustments. As early as possible in the semester, set up an appointment to meet with the instructor to discuss the academic adjustments specified in your accommodations letter as they pertain to my class.
What is a computer?

- Hard to define exactly
- The term "computer" originally referred to a person who did computations
- A computer is a machine that manipulates data according to a list of instructions
Kinds of Computers

**Supercomputers**
- Extremely expensive
- Often special purpose now

**Desktop Computers**
- For home and office use
- Some of the most powerful computers are clusters of desktops

**Laptop Computers**
- Hardly different from desktops now
- Focus on low power usage

**Tablet Computers**
- Taking the niche laptops once held
- Even lower power, usually no hard drive

**Embedded Computers**
- Tiny computers inside of watches, phones, toasters, cars, etc.
- More embedded computers than any other kind
## History of computers

<table>
<thead>
<tr>
<th><strong>Mechanical Calculation Devices (2400BC onward)</strong></th>
<th><img src="image1.png" alt="Image" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Aid to human calculation</td>
<td></td>
</tr>
<tr>
<td>• No stored program</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Mechanical Computers (1725 onward)</strong></th>
<th><img src="image2.png" alt="Image" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Punch card programming</td>
<td></td>
</tr>
<tr>
<td>• Serious limitations</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Early Electronic Computers (1941 onward)</strong></th>
<th><img src="image3.png" alt="Image" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>• General purpose, stored program computers</td>
<td></td>
</tr>
<tr>
<td>• Electronic, using vacuum tubes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Microprocessors (1970’s onward)</strong></th>
<th><img src="image4.png" alt="Image" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Succeeded transistors</td>
<td></td>
</tr>
<tr>
<td>• Now billions of computations per second at a nanometer scale</td>
<td></td>
</tr>
</tbody>
</table>
Hardware vs. software

- Hardware refers to physical parts of the computer
  - Processor
  - Memory
  - Hard disk
  - Monitor

- Software refers to the programs that run on it
  - Operating system (Windows, Mac OS, Linux, Unix)
  - Web browser (Internet Explorer, Safari, Firefox, Chrome)
  - Business applications (Word, PowerPoint)
  - Games
Von Neumann Architecture

- Basic layout of all modern computers
CPU

- The “brains” of the computer
- Fetches instructions and data from memory
- Performs computations on the data based on the instructions
- Can send results to I/O
- A modern CPU is made of electronic circuitry embedded in a small silicon chip
How fast are computers?
I typed this PowerPoint on a computer running at 2.6 GHz
That’s 2,600,000,000 cycles per second
Each cycle, your computer can do something like:
- Add
- Subtract
- Multiply
- (Usually not divide)
Moore’s Law

- “The density of transistors on a CPU doubles every 18 months”
- Historically, this has meant that CPU speeds have doubled every 18 months
- We can’t make things much faster because of heat and power
- We can still put more “stuff” into a CPU
- What do we do with that extra stuff?
Multicore

- Modern laptops and desktops are now almost all **multicore**
- Multicore means that each CPU actually has several independent processors called cores inside
- A CPU with 4 cores can actually be computing 4 different things at the same time
- Parallel processing
Parallel processing

- Works well for problems like washing loads of laundry in a laundromat

- But, if you have 3 loads of clothes, there is no way to wash them faster with 4 washers
Parallel processing works very poorly when different processors have to work on the same data and conflicts can happen.

Brain surgery with 100 surgeons is not 20 times faster than brain surgery with 5.

It’s not safer, either.
Memory

- Storage for all the data and instructions on your computer
- Modern computers store everything as binary digits (bits) which have a value of 0 or 1.

<table>
<thead>
<tr>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>8 bits</td>
</tr>
<tr>
<td>1 kilobyte (kb)</td>
<td>$2^{10}$ bytes</td>
</tr>
<tr>
<td>1 megabyte (mb)</td>
<td>$2^{20}$ bytes</td>
</tr>
<tr>
<td>1 gigabyte (gb)</td>
<td>$2^{30}$ bytes</td>
</tr>
<tr>
<td>1 terabyte (tb)</td>
<td>$2^{40}$ bytes</td>
</tr>
</tbody>
</table>
Memory

**Cache**
- Actually on the CPU
- Fast and expensive

**RAM**
- Primary memory for a desktop computer
- Pretty fast and relatively expensive

**Flash Drive**
- Faster than hard drives
- Seen on USB drives but SSDs are becoming common too

**Hard Drive**
- Secondary memory for a desktop computer
- Slow and cheap

**Optical Drive**
- Secondary memory that can usually only be written once
- Very slow and very cheap
Input/Output

Monitor
- Common visual output device

Speakers
- Common audio output device

Mouse
- Common input device

Keyboard
- Common input device
Computer Science
Now that we’ve (sort of) defined computers, what is computer science?
The study of information, computation, and solving problems with programs
Subfields:
- Theoretical computer science
- Programming languages and compilers
- Operating systems and networking
- Graphics
- Numerical computing
- Information storage, retrieval, and security
- Architecture and hardware
Computers are stupid, but fast
Programming is the process of giving them very detailed instructions about what to do
Usually, programming is done in a rigid, formalized language, like **Java**
English is insufficient:
- E.g., “Computer! Solve my relationship problems!”
- Writing a program to solve your relationship problems in **Java** would require you to be more detailed and explicit
Layers

- Computer science is built out of layers (like a burrito)
- No one can understand everything
- People tend to focus on a particular level
Layers of Abstraction

User

Application

Operating System

Hardware

We will program here
Upcoming
Next time...

- We will talk about software development
- We will introduce some Java basics
Reminders

- Read Chapter 1