Object-Oriented Design



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: The Point 105
- **Phone:** (614) 823-2944
- Office hours: MWF 9:00 10:15 a.m.,
 - **MWF** 3:00 5:00 p.m.,
 - **TR** 9:00 9:55 a.m.,
 - **TR** 2:00 5:00 p.m.,

and by appointment

Website:

http://faculty.otterbein.edu/wittman1/

Who are you?



Computer Science
Business Analytics
Mathematics
Physics
Undeclared

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?

Course Overview

Textbook

- Barry Wittman, Aditya Mathur, and Tim Korb
- Start Concurrent: An Introduction to Problem Solving in Java with a Focus on Concurrency
 - Available: <u>https://start-concurrent.github.io/</u>



A note about the book...

- The book's not bad
 - At least it's free
 - 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

Course focus

- Designing better, more reusable code
- More complex algorithms
- Testing code
- Features of Java we will focus on:
 - Interfaces
 - Inheritance
 - Exceptions
 - Graphical user interfaces (GUIs)
 - Recursion
 - File and network I/O
 - Java Collections Framework (JCF)
 - Regular expressions

More information

- For more information, visit the webpage: http://faculty.otterbein.edu/wittman1/comp2000
- 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/otterbein/spring2020/comp2000



Five projects

- 35% of your grade will be five equally weighted projects
- Each will focus on a different major area from the course:
 - Inheritance
 - GUI
 - Recursion
 - Linked lists (and networking)
 - Extensive library use
- You will work on each project in two-person teams



- All projects are done in teams of two
- You may pick your partners
 - But you have to have a different partner for each project!
 - Use Blackboard to form teams
- Projects must be uploaded to Blackboard
 (https://otterbein.blackboard.com/)

Turning in projects

- Projects must be uploaded to Blackboard before the deadline
- Do not put projects in your public directories
- Late projects will not be accepted
 - Exception: Each person will have 3 grace days
 - You can use these 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
 - If two people in a team don't have the same number of grace days, the number of days they will have available will be the maximum of those remaining for either teammate
- Assignments that don't compile get o points

Labs

In-class Programming Exercises

Labs

- 15% of your grade will be based around programming labs
- Labs are on Tuesdays and Thursdays
- Is of these labs will focus on the solution of a problem with a graded exercise
- Work should be done individually, but the goal is to learn, and I will help everyone
- The remaining lab days are to discuss course material and work on team projects
- You are expected to attend all lab days



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

- There will be two equally weighted in-class exams totaling 30% of your final grade
 - Exam 1: 02/10/2020
 - Exam 2: 03/23/2020
- The final exam will be worth 15% of your grade
 - Final: 10:15 a.m. 12:15 p.m. 04/27/2020

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	01/13/20	Java Recap	3 - 9	
2	01/20/20	Interfaces	10	MLK Day
3	01/27/20	Inheritance	11 and 17	
4	02/03/20	Exceptions	12	Project 1 Due
5	02/10/20	GUI	7 and 15	Exam 1
6	02/17/20	More GUI	15	
7	02/24/20	Recursion	18	Project 2 Due
	03/02/20	Spring Break		
8	03/09/20	Files	20	
9	03/16/20	Network I/O	21	Project 3 Due
10	03/23/20	Linked Lists	18	Exam 2
11	03/30/20	JCF	18	Project 4 Due
12	04/06/20	UML, design, and testing	16	Good Friday
13	04/13/20	Regular Expressions	Notes	
14	04/20/20	Review	All	Project 5 Due

Project schedule

- Project 1: 7% Tentatively due 02/07/2020
- Project 2: 7% Tentatively due o2/28/2020
- Project 3: 7% Tentatively due 03/20/2020
- Project 4: 7% Tentatively due 04/03/2020
- Project 5: 7% Tentatively due 04/24/2020

Policies

Grading breakdown



Grading scale

Α	93-100	B-	80-82	D+	67-69
A-	90-92	C+	77-79	D	60-66
B+	87-89	С	73-76	F	60-62
В	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 and labs 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 play with your phones
 - Don't use the computers in class unless specifically told to
 - No food or drink in the lab

Computer usage

- We will be doing a lot of work on the computers together
- 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 playing on your phones or using your computer for anything other than course exercises

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 Otterbein
- Refer to the syllabus for the school's policy
- Ask me if you have questions or concerns
- You are not allowed to look at another student's code, except for group members in group projects (and after the project is turned in)
- 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 within your team
 - If I can ascertain that code from one team's project appears in another team's project, both teams will score zero points
 - All students will also have a full letter grade reduction at the end of the semester

Learning Differences

If you have a documented learning difference please contact Kera McClain Manley, the Disability Services Coordinator, to arrange for whatever assistance you need. The Disability Services is located in Room #13 on the second floor of the Library in the Academic Support Center. You are welcome to consult with me privately to discuss your specific needs. For more information, contact Kera at kmanley@otterbein.edu, (614) 823-1618 or visit Disability Services.

How to Succeed in this Course

Bad students...

- Don't ask questions
- Don't come to office hours
- Don't ask for help
- Treat education as a passive experience
- Are happy when a class is easy
- In other words, they act as if college is high school

Good students...

- Ask questions
- Come to office hours
- Ask for help
- Actively pursue all the knowledge and skills they can
- Are angry when a class is easy

Flowcharts for COMP 2000

Flowchart for success:



Flowchart for failure:



Java Refresher



- Variables are used to store data in Java
- All variables must be declared:

int value;

When a variable is declared, it can also, optionally, be assigned at the same time:

double inches = 4.96;



 All variables have a type, which comes before the name of the variable in the declaration:

int value;

- Unlike dynamic languages like Python or JavaScript, the type of a variable never changes
- Types determine:
 - Legal values you can put in a variable (like integers or text)
 - Operations you can do on those variables (like addition or concatenation)
- Types come in two flavors: primitive types and reference types

Primitive types

Java has 8 primitive types:

	Туре	Bytes	Range	Purpose
	byte	1	-127 - 128	Tiny integers
	char	2	Many Unicode characters	Characters
Integer	short	2	-32,768 - 32,767	Small integers
integer	int	4	-2,147,483,648 - 2,147,483,647	Normal integers
	long	8	-9,223,372,036,854,775,808 – 9,223,372,036,854,775,807	Large integers
Floating point	float	4	(±) 1.4 × 10 ⁻⁴⁵ – 3.4 × 10 ³⁸	Low precision math
Floating-point	double	8	(±) 4.9 × 10 ⁻³²⁴ – 1.8 × 10 ³⁰⁸	High precision math
Other	boolean		true, false	Logic

Primitive type ranges

- Java has relatively strong typing
 - Understand why you're making a cast, and try not to make casts for no reason
- Remember that all the primitive numerical types in Java are signed
 - Strange things can happen

byte
$$x = -128;$$

$$x *= -1;$$

System.out.println(x); //output?

Special double values

- The integer types only have the values in the range listed
- No other values are possible, certainly not null
- The double and float types have a few special values:

float	double	Meaning
Float.NaN	Double.NaN	Not a representable number, such as the square root of a negative number
Float.NEGATIVE_INFINITY	Double.NEGATIVE_INFINITY	Too negative of a value, larger than can be represented, such as -5.0/0.0
Float.POSITIVE_INFINITY	Double.POSITIVE_INFINITY	Too positive of a value, larger than can be represented, such as 7.0/0.0

Basic math operations

- + adds
- subtracts
- * multiplies
- / divides (integer division for int type and fractional parts for double type)
- % finds the remainder
- Order of operations holds, and parentheses can be used to clarify

Shortcut notation

Java has a number of shortcuts for common operations

Shortcut	Meaning	Shortcut	Meaning
х += у;	$\mathbf{x} = \mathbf{x} + \mathbf{y};$	x++ ;	$\mathbf{x} = \mathbf{x} + 1$; (return old value)
х -= у;	$\mathbf{x} = \mathbf{x} - \mathbf{y};$	++x;	$\mathbf{x} = \mathbf{x} + 1$; (return new value)
х *= у;	$\mathbf{x} = \mathbf{x} * \mathbf{y};$	x ;	$\mathbf{x} = \mathbf{x} - 1$; (return old value)
х /= у;	$\mathbf{x} = \mathbf{x} / \mathbf{y};$	x ;	$\mathbf{x} = \mathbf{x} - 1;$ (return new value)
x %= y;	x = x % y;		

Shortcut notation

These shortcuts are *almost* the same as combinations of other operators, but they don't have the same type-checking:

And know what you're doing with ++:

int i = 0; i = i++; //legal but crazy i = ++i; //legal, crazy, different result

Upcoming

Next time...

- Come to lab tomorrow (even though we won't have a graded assignment) to refamiliarize yourself with Eclipse and make sure you can log in
- On Wednesday, we'll review:
 - Selection
 - Loops
 - Arrays
 - Static methods



Review Chapters 3 – 8 (except for 7)