# CS 2334:
# Programming Structures and Abstractions (aka Java 2)

Dr. Andrew H. Fagg

Schools of Computer Science and Biomedical Engineering

Teaching Assistants:

Daniel Brigance, Taner Davis, Prathyusha Kondapalli, Nicholas Sidwell

Many of the lecture materials used in this course are derived from the work of Dr. Amy McGovern and Dr. Deborah Trytten

# What do you want to do with your CS (or related) degree?

# Some Motivation …

- <u>Driverless cars:</u>
  - <u>http://www.youtube.com/watch?v=bp9KBrH8H04</u>
- Coding, the new literacy:
  - <u>https://www.youtube.com/watch?v=nKIu9yen5nc#t=25</u>

# Why CS is Fun (for me)

- Research areas: robotics (manipulation and assistive), computational neuroscience, machine learning, interactive art

- Interacting with individuals from different disciplines to solve hard computational problems
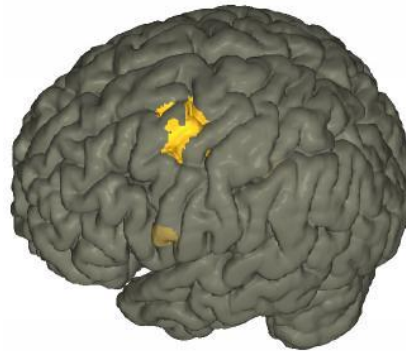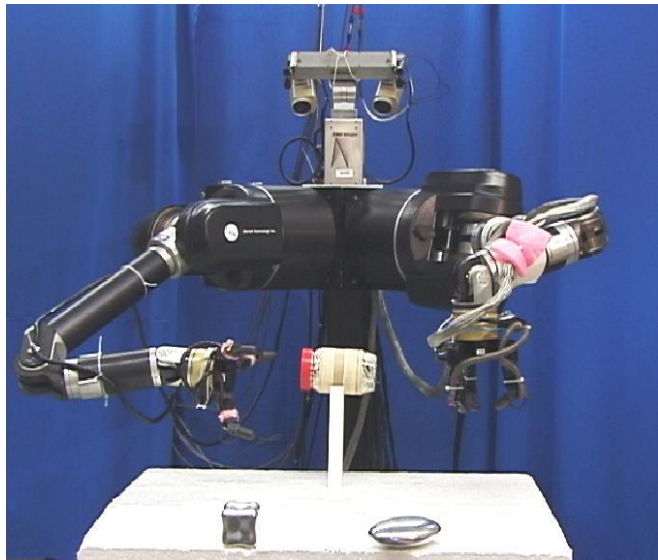
# This software stuff is hard … Why?

# This software stuff is hard … Why?

Complexity due to:

- Different types of data
- Users are diverse!
- Different use cases
- Different needs
- Code base gets large
- Multiple programmers
- Coordinating many activities at once

http://www.extremetech.com/extreme/118863-nevada-embraces-the-future-approves-self-driving-cars

# Why Should We Care?

and does it matter that we get it right?

# Why Should We Care?

Does it matter that we get it right?

- Correct and efficient implementation is important to our customers & employers

- Resources are often precious: e.g., data, people, and CPU

- Lives can be at stake (literally)

- We can change the world



**Helping children at risk for Cerebral Palsy learn how to crawl**

Photo credit: Hugh Scott

# This software stuff is hard …

How do we get a handle on it?

# Abstraction

# Abstraction

- Abstraction: the process of simplifying the representation or description of some entity
  - Keep the key pieces
  - Hide the extraneous details

- In software development: we use abstraction to temporarily hide details so that we can "get our mind" around the "big picture"

# Abstraction

Not just one level of abstraction possible: we can imagine multiple levels of abstraction, depending on what we are working on and what we need to communicate

# Course Coverage

- Abstraction and Object Oriented Programming
- Software development
  – Design
  – Implementation
  – Testing
  – Debugging
- Ethics in computer science

# Design

Design: the process of assessing the requirements of a software system and planning a solution

- What are the inputs and outputs?

- What happens in between and how?

- How do we know when our implementation is correct?

Abstraction is key for many of these steps

# Implementation

- Connecting our design and our solution

- Maintaining a separation of the logic of our solution from the implementation

- Tools that help us to manage our abstractions

# Testing and Debugging

- Testing procedures are designed (often ahead of time)

- Testing procedures for different pieces of the code base

- Tools that allow us to analyze what our code is doing and what it is "thinking"

- Isolation of "buggy" code

# Ethics in Computer Science

Processes for detecting and analyzing ethical questions that can arise in the computing solutions that we develop

- Privacy

- Intellectual property

# My Assumptions About You

- At least one introductory course in programming

- Experience with java, including:
  - Control structures: if-then-else, while, for, switch
  - Basic data types: integers, floats, chars, strings
  - Exposure to java objects

- Experience with writing and debugging your own programs

# ABET outcomes

- B:  An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution

- C:  An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs

- K:  An ability to apply design and development principles in the construction of software systems of varying complexity

- E: An understanding of professional, ethical, legal, security, and social issues and responsibilities.

# Course Details

# Course Information

- Instructor: Dr. Andrew H. Fagg

- Class Location: Dale Hall 206

- Required Resources:
  - Zyante Java programming
  - The **Fourth** Edition of *A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet*
  - *Top Hat*

- Prerequisites: 1323 and Mathematics 1523 or higher, both with a grade of C or higher

- Course web page:

  http://www.cs.ou.edu/~fagg/classes/cs2334/

# How to Find Me

- Office: DEH 243
- Office Hours:
  - M: 11:45-1:15
  - W: 11-12:30
  - Also by appointment
- Contact information: andrewhfagg@gmail.com

# How to find the TAs

All office hours in DEH 115 (computer lab)

Daniel Brigance        brigance@ou.edu
Taner Davis            taner@ou.edu
Prathyusha Kondapalli  prathyusha_kondapalli@ou.edu
Nicholas Sidwell       nicholas.sidwell@ou.edu

Appointments can also be made
All of us can be reached simultaneously: cs2334@cs.ou.edu

# Course policies

Due Dates:

- Homework (Zyante exercises): start of class on the day assigned

- Projects: start of class (1:29pm) on the due date
  - Project 5 is different

- Labs: end of the day on Friday (11:59pm)

# Course policies

Late policy:

- Homework and labs cannot be turned in late for credit

- A project may be turned in late for a penalty:
  - 0 - 24 hours:10%;
  - 24 - 48 hours: 20%

- In-class exercises (Top Hat) and exams must be completed in class

# Laptop Policy

- Labs: laptops are required

- Class:

  - May be used to program along with the rest of the class or to take notes

  - May want to use for Top Hat

  *If you are using your laptop to in a way to distract people around you in class, you will be asked to leave.*

# Tools

- Canvas: grade book, announcements, discussion, calendar
- Main course web site: http://cs.ou.edu/~fagg/classes/cs2334
- Catme: project group assignment (email coming)
- Zyante: on-line text book (details coming)
- Top Hat: in-class exercises and discussion (email sent)
- Eclipse: integrated development environment
- Web-Cat: program submission and grading

# Grading

- 5 Projects: 25%

- 15 Labs: 15% (keep 14 highest; must keep labs 14 & 15)

- Exams: 40% (two midterms and a final)

- Homework: 10% (exercises in the Zyante readings; keep N-1 highest)

- In-class exercises: 10% (Top Hat; keep M-1 highest)

Official grades will be posted in Canvas

# Laboratory Assignments

- Attend the lab session in which you have enrolled

- Lab sessions are 110 minutes long
  - Short lecture and introduction to the lab assignment
  - Time to work on the lab itself and get help from the teaching assistants

- Labs 1-13 are due on Friday

- Labs 14-15 are due at the end of the session

# Projects

- 5 two-week long projects over the semester
- You will need this time
- Projects are done in assigned pairs

- Grading:
  - Sign up for a code review time slot
  - Both group members must be present at the review
  - Both must be ready to answer any questions about the code

# Grading questions

- The item should be first brought to the person who graded it

- All grading questions must be brought to our attention within <span style="color:red">one week</span> of when the item was returned

- Check your grades on Canvas

- Exams: once the exam leaves the room that I am in, the grade is final

# Getting the most out of class

- Read materials ahead of time
- Ask questions (in person or in Top Hat)
- Learn names of your fellow students (and use them)
- Participate in class discussions
- Participate in your group discussions
- Attend class regularly
- At the end of the semester, I should know your name

# Appropriate Classroom Conduct

One rule: <span style="color:red">Respect</span>

- Yourself

- Your peers

- The teaching team
  - Keep in mind: we are human, too, and we have many obligations

# Inappropriate Classroom Conduct

- Allowing a cell phone or pager to repeatedly beep audibly.

- Playing music or computer games during class in such a way that they are visible or audible to other class members.

- Exhibiting erratic or irrational behavior.

- Behavior that distracts the class from the subject matter or discussion.

- Making physical or verbal threats to a faculty member, teaching assistant, or class member.

- Refusal to comply with faculty direction.

# Classroom Conduct

- In the case of disruptive behavior, we will ask that you leave the classroom and I may charge you with a violation of the Student Code of Responsibilities and Conduct.

- If you have repeated disruptive issues, I will seek to withdraw you from the class.

# Proper Academic Conduct

Laboratory assignments, homework assignments, in-class exercises and exams:

- All work must be your own: no looking at or copying solutions from other students or from the net

- General discussion is OK (e.g., the fundamental skills that we are learning)

- When in doubt: ask me or a TA

# Proper Academic Conduct

Projects:

- All work must be that of your group: no looking at, discussing or copying solutions from other groups or from the net

- General discussion across groups is OK

- Group members must contribute equally to each project

# Proper Academic Conduct

- Sharing solutions is punished to the same degree as receiving solutions

- Make sure that your computer and account are properly protected. Use a good password

- Do not give out access to your account or your computer system

- Do not leave printouts or mobile drives around a laboratory where others might access them

# Proper Academic Conduct

Programming projects will be checked by software designed to detect collaboration. This software is extremely effective and has withstood repeated reviews by the campus judicial processes.

# What happens if you cheat…

- Upon the first documented occurrence of collaborative work, I will report the academic misconduct to the Campus Judicial Coordinator. The procedure to be followed is documented in the University of Oklahoma Academic Misconduct Code

  - http://integrity.ou.edu/files/Academic_Misconduct_Code.pdf

- The appeals process for both admonitions and full complaints is described at:

  - http://integrity.ou.edu/

# This Week…

- Reading: email etiquette

- Next time: 1323 review

- Lab: JDK, Compiling, Javadoc, Eclipse, Strings
  - Exercise is due on Friday evening