# Csci 1933: Introduction to Algorithms and Data Structures

Fall 2017

# Course overview

- Main goal:  learn about data structures and associated algorithms that are fundamental to all of computer science
- Learned in context of Java, modern object-oriented programming language (more later on what this means)
- Exposure to practical approaches and tools for designing/implementing good software

# Course staff introductions

- Me: Chad Myers
  - Grew up in ND
  - Grad. school: Princeton, Computer Science
  - At UofM since Jan. 2008
  - Research specialty: computational biology (applying CS to large biological datasets, e.g. human genome)
  - Hobbies/interests: snow skiing, water sports (skiing, wakeboarding/surfing), outdoor stuff, football, making sense of data
- chadm@umn.edu
- Office hours:

  Molecular and Cellular Biology Building (MCB)

  Wednesdays 3:30-5pm, Thursdays 3-4:30pm (and by appointment)

My genome

- 1.4 billion reads- 140 billion base pairs of sequence data
- ~90 GB of data
- 3.2 billion total bases (A,C,T,G)
- 3.2 million bases make me unique (~1/1000 bases are different)

# Computational Systems Biology Lab

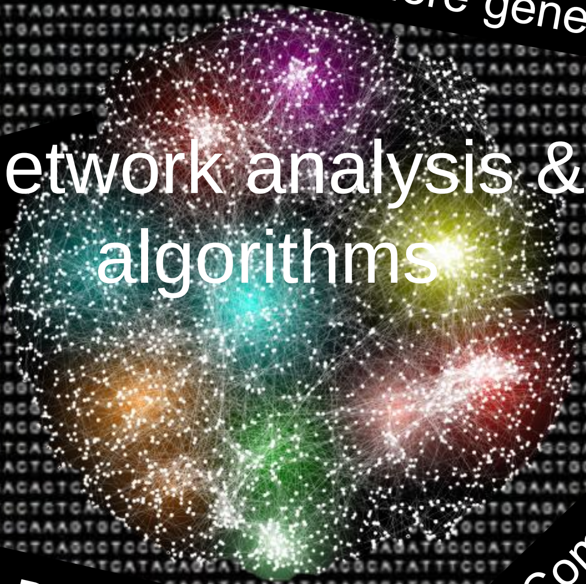
http://csbio.cs.umn.edu/

Email me if you're interested in learning more!

How do gene mutations interact to cause disease?

Inferring/analyzing "logic circuits" that control when/where genes are expressed

How do drugs interact with genes?

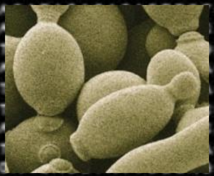


Network analysis & algorithms

Highly collaborative

Complex, real-world problems with potential for high impact

Machine learning/data mining on BIG "fresh" data



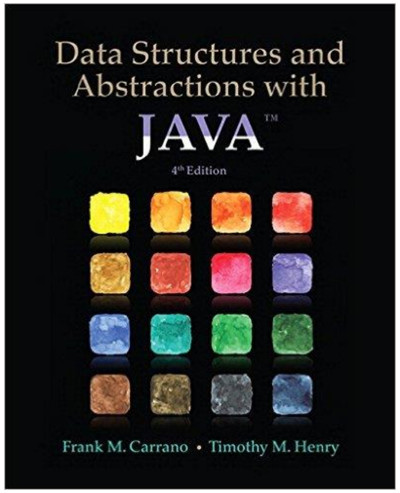
# Course staff introductions

- TAs
  - Allison Miller (mill7079_at_umn.edu), Lab Section: 5, 8
  - Andrew Gilbert (gilbe918_at_umn.edu), Lab Section: 3, 5
  - Aronee Dasgupta (arone008_at_umn.edu), Lab Section: TBD
  - Erica Loukusa (louku004_at_umn.edu), Lab Section: 2, 4
  - Henry Hoang (hoang159_at_umn.edu), Lab Section:  4, 8
  - Jason Zhang (zhan3507_at_umn.edu), Lab Section: 2, 8
  - Levi Mathwig (mathw011_at_umn.edu), Lab Section:  6, 7
  - Nate Larson (lars5228_at_umn.edu), Lab Section: 6
  - Niti Halakatti (halak004_at_umn.edu), Lab Section: TBD
  - Owen Houghton (hough083_at_umn.edu), Lab Section: 5, 6
  - Rahul Parhi (parhi003_at_umn.edu), Lab Section: 4, 7
  - Ryan Zoeller (zoell031_at_umn.edu), Lab Section: 2, 3
  - Sruti Paladugu (palad007_at_umn.edu), Lab Section: 3, 7

Group email: csci1933-help@umn.edu

# Course resources

- Textbook
  - Frank M. Carrano and Timothy M. Henry, Data Structures and Abstractions with Java, 4th Edition. Pearson Education, 2015.
    (we will be using several sections of the textbook, with some jumping around)


Data Structures and Abstractions with
JAVA™
4th Edition
Frank M. Carrano • Timothy M. Henry

- Course Moodle site: https://ay17.moodle.umn.edu/course/view.php?id=6173

- Lecture notes—text version posted online after the lectures  (mostly text notes, example code, a few powerpoints)

- Other good textbooks/online resources:
  - Oracle's tutorials: http://docs.oracle.com/javase/tutorial/java/index.html
  - O'Reilly's "Learning Java"
  - See "Resources" on course Moodle site

# Course logistics/policies

- See your syllabus, which is also posted on the Moodle site:

https://ay17.moodle.umn.edu/mod/resource/view.php?id=518844

# Important announcement about labs

- **Lab sessions start next Tuesday, 9/12**

  Section 002: Tuesdays 8:00-9:55am, Keller Hall 1-262
  Section 003: Tuesdays 10:10am-12:05pm, Keller Hall 1-262
  Section 004: Tuesdays 12:20pm-2:15pm, Keller Hall 1-262
  Section 005: Tuesdays 2:30-4:25pm, Keller Hall 1-262
  Section 006: Tuesdays 4:40-6:35pm, Keller Hall 1-262
  Section 007: Tuesdays 6:50-8:45pm, Keller Hall 1-262
  Section 008: Wednesdays 8:00-9:55am, Keller Hall 1-262

- **You should get a CSE Labs account before your lab next week (Tues/Wed) if you don’t have one already:**

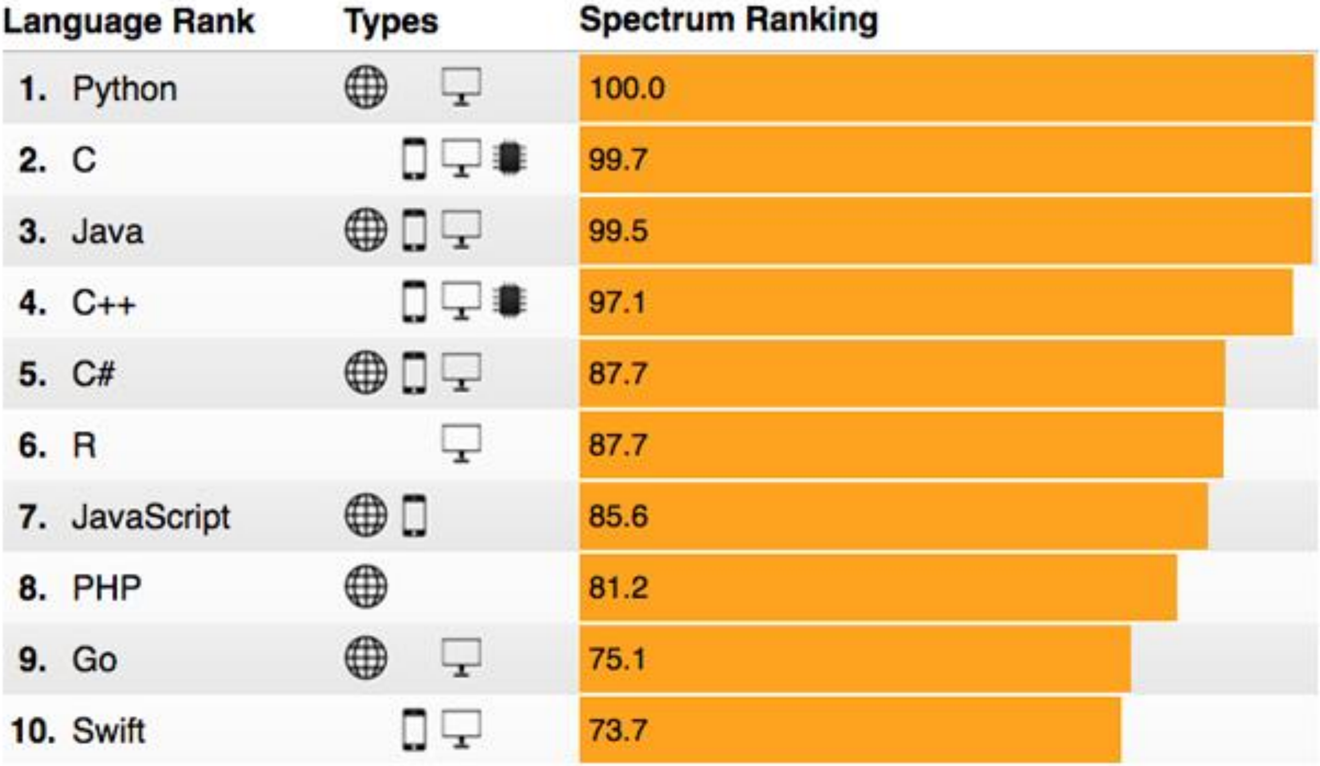
  **https://wwws.cs.umn.edu/account-management/**

# Short introduction to course material

# Course goals

- Learn about data structures and associated algorithms that are fundamental to all of computer science
- Learn to solve complex problems, using the appropriate data structures/algorithms, and be able to justify your design/critique others’ designs
- Learn Java, a modern object-oriented programming language
- Learn basic analysis of complexity of algorithms
- Learn practical approaches and tools for designing good software (e.g. IntelliJ)
- Learn to work in small teams to solve complex problems

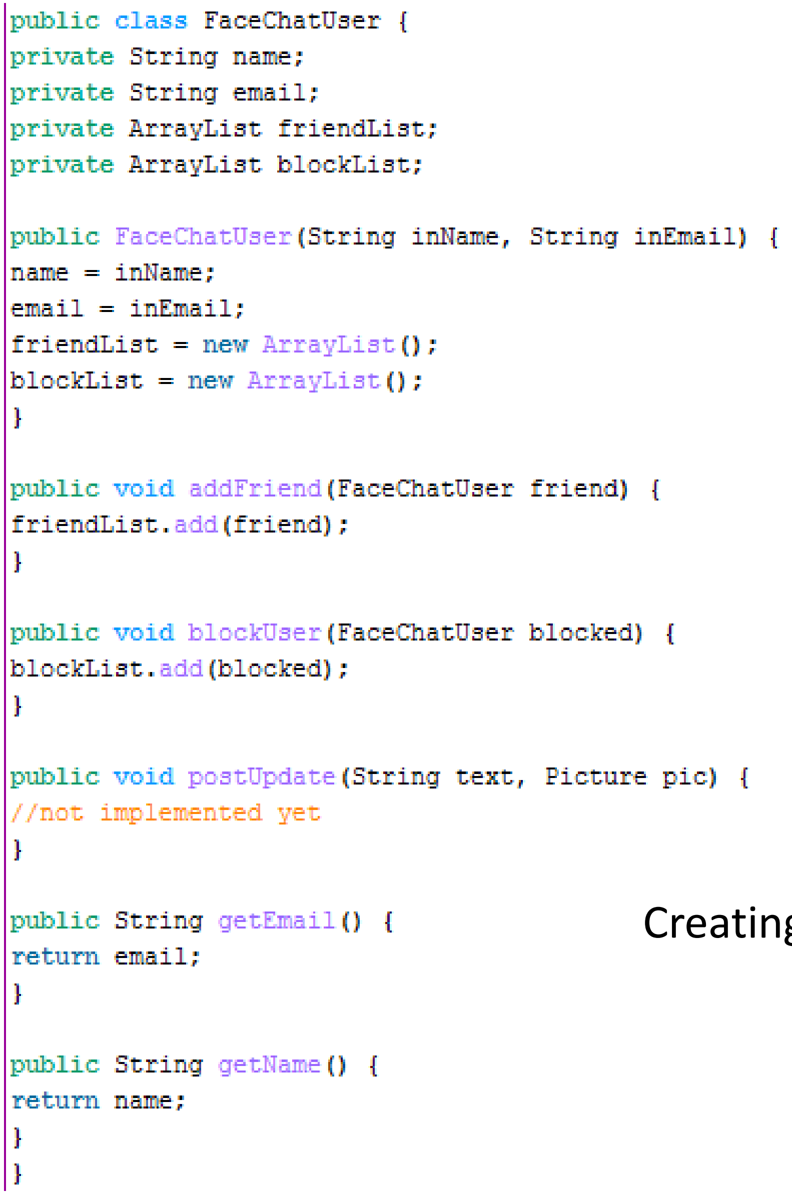
# Java is not the main emphasis of the course, but it is useful in the real world



| Language Rank | Types | Spectrum Ranking |
|---|---|---|
| 1. Python | 🌐 🖥 | 100.0 |
| 2. C | 📱 🖥 ▣ | 99.7 |
| 3. Java | 🌐 📱 🖥 | 99.5 |
| 4. C++ | 📱 🖥 ▣ | 97.1 |
| 5. C# | 🌐 📱 🖥 | 87.7 |
| 6. R | 🖥 | 87.7 |
| 7. JavaScript | 🌐 📱 | 85.6 |
| 8. PHP | 🌐 | 81.2 |
| 9. Go | 🌐 🖥 | 75.1 |
| 10. Swift | 📱 🖥 | 73.7 |

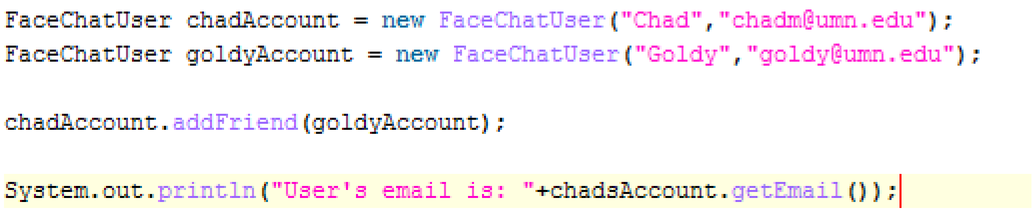
https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages

# An example Java class



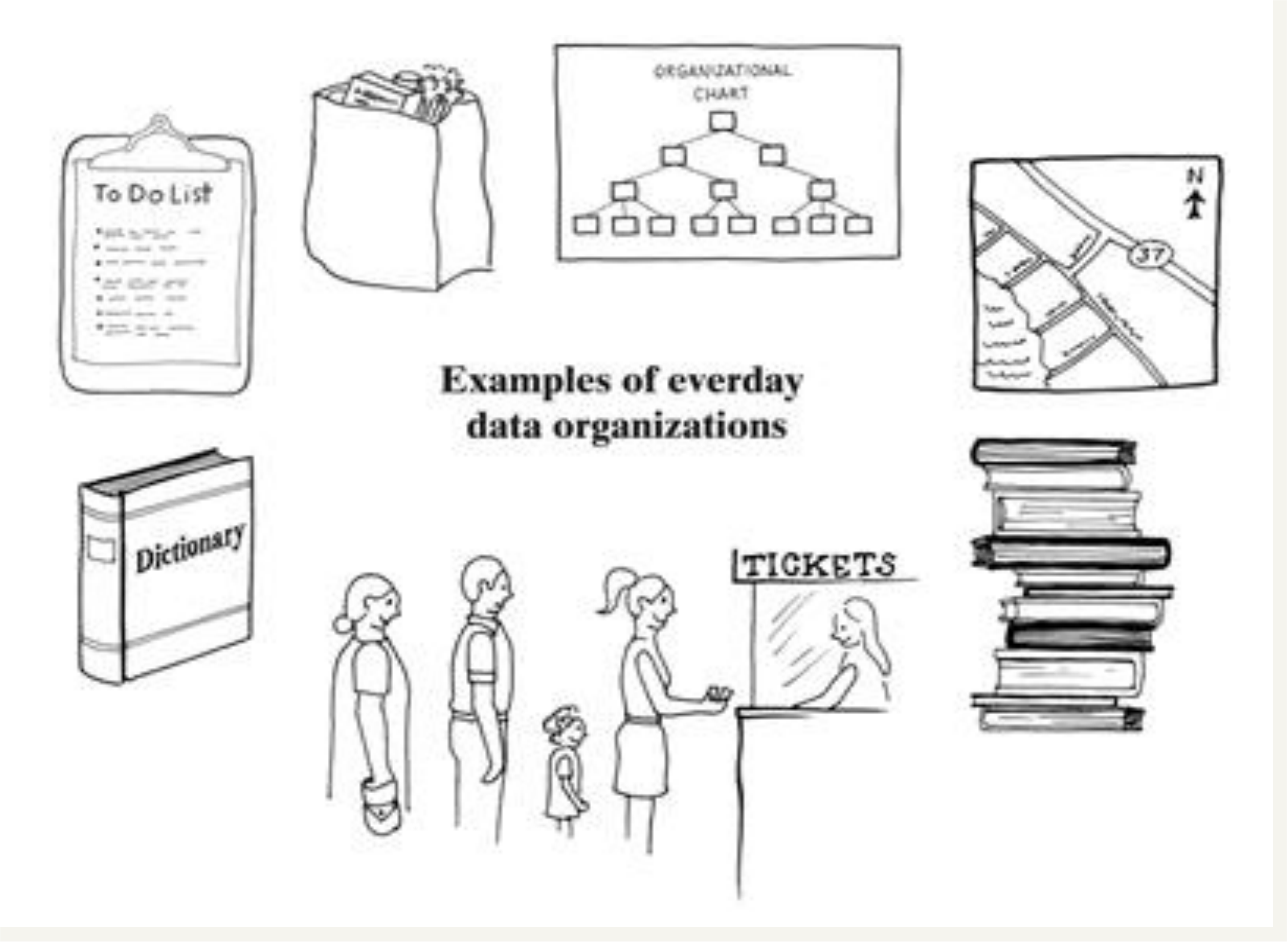
```java
public class FaceChatUser {
private String name;
private String email;
private ArrayList friendList;
private ArrayList blockList;

public FaceChatUser(String inName, String inEmail) {
name = inName;
email = inEmail;
friendList = new ArrayList();
blockList = new ArrayList();
}

public void addFriend(FaceChatUser friend) {
friendList.add(friend);
}

public void blockUser(FaceChatUser blocked) {
blockList.add(blocked);
}

public void postUpdate(String text, Picture pic) {
//not implemented yet
}

public String getEmail() {
return email;
}

public String getName() {
return name;
}
}
```

Creating instances of the class (objects):



```java
FaceChatUser chadAccount = new FaceChatUser("Chad","chadm@umn.edu");
FaceChatUser goldyAccount = new FaceChatUser("Goldy","goldy@umn.edu");

chadAccount.addFriend(goldyAccount);

System.out.println("User's email is: "+chadsAccount.getEmail());
```

# Organization is everywhere



Carrano, Data Structures and Abstractions with Java

# Next time…

- More introduction to object-oriented design: defining classes, methods, object-oriented design
- Basics of Java programming
- Java programming environment (IntelliJ)

# Be sure to sign up for your lab account!

- **You should get a CSE Labs account before your lab next week (Tues/Wed) if you don’t have one already:**

**https://wwws.cs.umn.edu/account-management/**