Today you'll need computer science

“Our species needs, and deserves, a citizenry with minds wide awake and a basic understanding of how the world works.”

-Carl Sagan
Who has taken Compsci 201?

Who am I?

Who Are You?

What year are you at Duke? (339 responses)


Who Are You?

What is the OS running on your laptop? (339 responses)
Who Are You?

How comfortable are you with Python programming? (329 responses)

- haven't read or written any Python: 44%
- have read, but never written Python: 9%
- have some experience with Python, e.g., have taken CompSci 101: 34.2%
- have more than some experience, have written lots of Python: 8.2%

Who Are You?

How comfortable are you with Java programming? (329 responses)

- haven't read or written any Java: 41.6%
- have read, but never written Java: 16.5%
- have some experience with Java, e.g., have taken AP Computer Science: 35.7%
- have more than some experience, have written lots of Java: 6.8%

Who Are You?

How comfortable are you with Matlab programming? (329 responses)

- haven't read or written any Matlab: 51.6%
- have read, but never written Matlab: 10.3%
- have some experience with Matlab, e.g., have taken EGR course with Matlab: 22.2%
- have more than some experience, have written lots of Matlab: 16.1%

Who Are You?

What kind of smartphone do you have? (329 responses)

- Don't have a smartphone: 78.8%
- iPhone: 21.2%
Who Are You?

Which of the following best describes why you're taking CompSci 201

- 19.9%
- 25.7%
- 0.0%

Considering a compsci major
Requirements, but major is doubtful
Want to learn more, don’t have to take this, major is nontotally
I have a friend taking it

Computer Science

- What is Computer Science?

- Where do we find an authoritative answer?
  - How do you answer questions? Where do you look?

- How is this relevant to CompSci 201?
  - What's the best approach to ...
  - How do you effectively solve this problem?

What is Computer Science?

What is it that distinguishes it from the separate subjects with which it is related? What is the linking thread which gathers these disparate branches into a single discipline? My answer to these questions is simple --- it is the art of programming a computer. It is the art of designing efficient and elegant methods of getting a computer to solve problems, theoretical or practical, small or large, simple or complex.

C.A.R. (Tony) Hoare

Programming != Computer Science

- What is the nature of intelligence? How can one predict the performance of a complex system? What is the nature of human cognition? Does the natural world 'compute'?

- It is the interplay between such fundamental challenges and the human condition that makes computer science so interesting. The results from even the most esoteric computer science research programs often have widespread practical impact. Computer security depends upon the innovations in mathematics. Your Google search for a friend depends on state-of-the-art distributed computing systems, algorithms, and artificial intelligence.

Efficient design, programs, code
Tradeoffs, Tradeoffs, Tradeoffs

Simple, elegant, quick, efficient. Don't worry about getting it right the first time. Plan for an iterative approach

Tension between generality, simplicity, elegance, effectiveness

Runtime, storage/space, development time, platform: basics of standard data structures and algorithms

Engineer, scientist: what toolkits do you bring to programming? Mathematics, design patterns, libraries --- standard and others...

Computer Science

● In 2015 we see
  ➢ Snapchat share 8,796 photos/second, 760 million/day
  ➢ Whatsapp has 700 million/day
  ➢ Facebook quoted at 70 million/day (2016: 350 million/day)

● Where is the computer science in these statistics?
  ➢ Where can you see photos?
  ➢ When can you see photos?
  ➢ Who can see photos?

Goals for the Course

● Given a problem statement & a real data source, design, develop, debug, and test a Java program that uses appropriate standard libraries to efficiently solve the problem.

● Write programs that effectively implement and use data structures such as: arrays, maps, linked lists, stacks, queues, trees, and graphs.

● Evaluate the time and space complexity of algorithms, especially algorithms that scale, using empirical or mathematical analysis.

● Apply basic object-oriented design and programming principles in developing software

PFTD and toward the week

● Be able to articulate why 201 is the right course for you, in terms of being able to complete it with understanding

● Be able to explain what work is expected, collaboration policies, exams, recitations, assignments, APTs

● Be able to read some Java programs and to analyze them by applying your knowledge of programming to Java programs

● Know what work you should complete before recitation on Monday and before end of next week
Latanya Sweeney

I am a computer scientist with a long history of weaving technology and policy together to remove stakeholder barriers to technology adoption. My focus is on "computational policy" and I term myself a "computer (cross) policy" scientist. I have enjoyed success at creating technology that weaves with policy to resolve real-world technology-privacy clashes.

http://latanyasweeney.org/

Identify 87% of US population using (dob, zip, gender). Prof. Government and Technology @ Harvard, instrumental in HIPAA because if de-identification work.

Former CTO of the FTC

Course Overview

- There are details, see the course web page
  - When are the midterm and final? What is policy for resources used on exams?
  - APTS: Algorithmic Problem-solving and Testing
    - Weekly small programming assignments, tested online
  - Programming assignments: solo, paired, ...
  - New this fall: WoTo – working together

- Why should you come to class?
  - Meet people, learn things, participate in a community
  - Provide help, get help, wonder, dance, think
  - Build community, answer questions

- Why is this course so great?
  - Because you're in it

What's in Compsci 201?

- Understanding tradeoffs: reasoning, analyzing, describing...
  - Algorithms
  - Data Structures
  - Programming
  - Design

- Programming using Java
  - Tools: Eclipse, JDK, Libraries, ...
  - Ideas: Design Patterns, OOP, Agile programming, ...
  - Engineering and analyzing designs and programs
  - Using mathematical and scientific techniques
  - Scaling solutions

Language  

http://www.java.com/en/about/
Problem Solving and Programming

- How many words are in a file?
  - What’s a word?
  - What’s a file?
  - How do we solve this: simply, quickly, ...?
    - What’s the best we can do? Constraints?

- How many different/unique words are in a file?
  - How is this related to previous task?
- What word occurs most often in ...
  - From Statistical analysis to Computational Bio

- How many words do two files have in common?
  - Spell-checking, Google did you mean ..?

Fast, cheap, out-of-control?


```java
import java.util.*;
import java.io.*;
public class StaticUnique {
    public static void main(String[] args)
        throws FileNotFoundException{
        Scanner s = new Scanner(new File("/data/kjv10.txt"));
        s.useDelimiter("\Z");
        String all = s.next();
        String[] words = all.split("\s+");
        TreeSet<String> set = new TreeSet<String>();
        set.addAll(Arrays.asList(words));
        System.out.printf("total #: %d, unique #: %d\n",
                words.length, set.size());
    }
}
```

Finding Maximally Occurring Word

- Tradeoffs with three different methods
  - Not expected to understand these
  - Goal: all familiar at end of next week
  - git.cs, classwork project


How fast is fast? How cheap is cheap?

- How do we measure how fast the code/design is?
  - Can we implement this design in C++?
  - Can we implement this in Ruby?

- We want a measure that’s independent of language?
  - What are we measuring?
  - How do we express answer?
  - Units? Best case? Average? Worst?

- What is answer using recognized terminology?
Some Java Vocabulary and Concepts

- **Java is a (statically) typed language**
  - Every variable/object has a type that's defined at compile time (compare Python)
  - More typing, but more errors caught before runtime

- **Java is an object-oriented language**
  - All code is in a class and all code inside function/method
  - Classes are also types, so variables/objects extendible
  - We will see interfaces and inheritance later
  - Classes have instance variables, methods local variables

- **Java has different 'types' primitive and Object/class. This is for efficiency, but it's sometimes confusing**

More Java Vocabulary and Concepts

- **Java has a huge standard library**
  - Organized in packages: `java.lang, java.util, javax.swing, ...`
  - API browseable online, but Eclipse IDE helps a lot

- **Java methods have different kinds of access inter/intra class**
  - Public methods ...
  - Private methods ...
  - Protected and Package methods ...

- **Primitive types (int, char, double, boolean) are not objects but everything else is literally an instance of class Object**
  - `foo.callMe();`

Solving problems, writing code

- **APT: AccessLevel**
  - Understand the problem, know how to solve an instance
  - Ideas? Caveats?

- **Writing code to implement proposed solution**
  - Will it run? In time? Constraints? Look before you code
  - How will we test the solution? When to start testing?

- **What's the green dance and when do we do it?**
  - Satisfaction of finishing something
  - Knowing when to stop when you're not making progress
  - Leveraging community wisdom

Analysis of StaticSimpleWords.java

- **How are variables used?**
  - What types are there, definitions, re-used variables,
  - What is a File, a Scanner, a String[], a HashSet, a ...

- **How are strings and values printed?**
  - What is System.out? How is it used?

- **Why is there a FileNotFoundException thrown from main?**
  - What is an exception, how do we handle them?

- **How are statements organized, how is whitespace used?**
Who takes Compsci 201 Now?

Duke Connection: Fred Brooks '53

- **What Would FB Say?**
  "The most important single decision I ever made was to change the IBM 360 series from a 6-bit byte to an 8-bit byte, thereby enabling the use of lowercase letters. That change propagated everywhere."

- **Fred Brooks** by Copyright owned by SD&M (www.sdm.de) - Request for picture sent by email to Fred Brooks by uploader (Mark Pellegino) and contact information for Carola Lauber at SD&M, who gave copyright permission. Licensed under CC BY-SA 3.0 via Wikimedia Commons - https://commons.wikimedia.org/wiki/File:Fred_Brooks.jpg#/media/File:Fred_Brooks.jpg

Why is programming fun?  

- **Fred Brooks**
  - First is the sheer joy of making things
  - Second is the pleasure of making things that are useful
  - Third is the fascination of fashioning complex puzzle-like objects of interlocking moving parts
  - Fourth is the joy of always learning
  - Finally, there is the delight of working in such a tractable medium. The programmer, like the poet, works only slightly removed from pure thought-stuff."