What is Computing? Informatics?

- What is computer science, what is its potential?
  - What can we do with computers in our lives?
  - What can we do with computing for society?
  - Will networks transform thinking/knowing/doing?
  - Society affecting and affected by computing?
  - Changes in science: biology, physics, chemistry, ...

- Privileges and opportunities available if you know code
  - Writing and reading code, understanding algorithms
  - Majestic, magical, mathematical, mysterious, ...

Computer Science: Duke Connection

- What we tell you it is
  - A bunch of courses useful in some majors

- What you want it to be or imagine it to be
  - Independent study, new courses, interdepartmental major

- What will it be in one year or two?
  - New courses, new professors, new majors, ...

- What is it outside of Duke?
  - Similar but different!

Computer Science: Duke Connection


Fundamental Compsci Concepts

- “Mathematics is the Queen of the Sciences”
  - Carl Friedrich Gauss

- What is Computer Science?
  - Why study it, what is it, why is it interesting (or not)?

- Historically
  - What can we program: at all, efficiently, optimally

- Present
  - Lots of data, lots of connectivity, lots of inferences

- Future
  - Where do we go from here?
What can be programmed?

- **What class of problems can be solved?**
  - Hadoop, Intel i7, Mac, Windows 7, Android, ...
  - Alan Turing contributions
    - Halting problem, Church-Turing thesis

- **What class of problems can be solved efficiently?**
  - Problems with no practical solution
    - What does practical mean?
  - We can’t find a practical solution
    - Solving one solves them all
    - Would you rather be rich or famous?

Schedule students, minimize conflicts

- **Given student requests, available teachers**
  - Write a program that schedules classes
  - Minimize conflicts

- **Add a GUI too**
  - Web interface
  - ...
  - ...

Still another scenario, is this better?

- I can’t write this program because I’m too dumb

Entscheidungsproblem

- **What can we program?**
  - What kind of computer?

- **What can’t we program?**
  - Can’t we try harder?

- **Can we write a program that will determine if any program P will halt when run on input S?**
  - Input to halt: P and S
  - Output: yes/no halts
Good sites: http://del.icio.us/

- What is social bookmarking?
  - Why is del.icio.us interesting?
  - Who posts, who visits?

- What about a website of interesting websites?
  - What would you expect to find there?
  - Would the site list itself?

- What about sites that list/link to themselves?
  - What about a site with all sites that list themselves?

Bad sites: http://haz.ardo.us

- Sites listing bad sites (don’t visit them?)
  - Where would this be useful?
  - What about censorship (internationally?)

- What about sites that list/link to themselves?
  - Is haz.ardo.us there?

- Website of all the sites that don’t list themselves?
  - Is notlisted.com listed on notlisted.com?

halting module/problem: writing doesHalt

```python
""
    function doesHalt returns True if program
    halts when run on input, and False if program
    doesn't halt (infinite loop)
    ""
    def doesHalt(programme,input):
        #code here
        name = "ActivateGame.py"
        data = "input.txt"
        if doesHalt(name,data): print "program ended!"

- A Python interpreter reads other programs as input
  - Can a word counting program count its own words?
- The doesHalt method might simulate, analyze, ...
  - One program/function that works for any program/input

How to tell if X stops/halts on Y

import halting
def runHalt():
    prog = "ActivateGame.py"
    input = "["abc", "def", "hij"]"
    if halting.doesHalt(prog,input):
        print prog,"stops"
    else:
        print prog,"loops 4ever"

- Can user enter name of program, X? Input, Y?
  - What’s the problem with this program?
Consider this module *Confuse.py*

```python
import halting
print "enter name of program",
prog = raw_input()
if halting.doesHalt(prog,prog):
    while True:
        pass
    print "finished"
```

- We want to show writing `doesHalt` is impossible
  - Proof by contradiction:
  - Assume possible, show impossible situation results

- Can a program read a program? Itself?

---

**Travelling Salesperson**

- Visit every city exactly once
- Minimize cost of travel or distance
- Is there a tour for under $2,000? less than 6,000 miles?
- Is close good enough?
  - Within 10% of optimal
  - Within 50% of optimal
  - ...

Try all paths, from every starting point -- how long does this take?

a, b, c, d, e, f, g
b, a, c, d, e, f, g ...

---

**Theory and Practice**


- How do we create an Ajax-based solution to the data-based back website we're building to do online course evaluations?

---

**Are hard problems easy? Clay Prize**

- $P = NP$
  - $P$ means solvable in polynomial time
    - Difference between $N$, $N^2$, $N^{10}$?
  - NP means non-deterministic, polynomial time
    - guess a solution and verify it efficiently

- Question: $P = NP$?
  - if yes, a whole class of difficult problems, the NP-complete problems, can be solved efficiently
  - if no, no hard problems can be solved efficiently
  - showing the first problem was NP complete was an exercise in intellectual bootstrapping, satisfiability/Cook/(1971)
Theory and Practice

- **Number theory: pure mathematics**
  - How many prime numbers are there?
  - How do we factor?
  - How do we determine primeness?

- **Computer Science**
  - Primality is *easy*.
  - Factoring is *hard*.
  - Encryption is possible.

 Wikileaks, PGP, PKI, verification

- [http://cryptome.org/0001/wikileaks-keys/wikileaks-keys.htm](http://cryptome.org/0001/wikileaks-keys/wikileaks-keys.htm)

- **Where are wikileaks servers and how to find them?**
  - What if they’re taken down
  - Where is information
  - What about imposters or verification?

- **File x distributed**
  - Download
  - Verify integrity and source!

A Rose by any other name... C or Java?

- **Why do we use [Python | Java] in courses?**
  - [is | is not] Object oriented
  - Large collection of libraries
  - Safe for advanced programming and beginners
  - Harder to shoot ourselves in the foot

- **Why don’t we use C++ (or C)?**
  - Standard libraries weak or non-existent (comparatively)
  - Easy to make mistakes when beginning
  - No GUIs, complicated compilation model
  - What about other languages?

Why do we learn other languages?

- **Perl, Python, PHP, Ruby, C, C++, Java, Scheme, ML,**
  - Can we do something different in one language?
    - Depends on what different means.
    - In theory: no; in practice: yes
  - What languages do you know? All of them.
  - In what languages are you fluent? None of them

- **In later courses why do we use C or C++?**
  - Closer to the machine, understand abstractions at many levels
  - Some problems are better suited to one language
    - Writing an operating system? Linux?
Unique words in Java

```java
import java.util.*;
import java.io.*;
public class Unique {
    public static void main(String[] args)
        throws IOException{
        Scanner scan =
            new Scanner(new File("/data/melville.txt"));
        TreeSet<String> set = new TreeSet<String>();
        while (scan.hasNext()){
            String str = scan.next();
            set.add(str);
        }
        for(String s : set){
            System.out.println(s);
        }
    }
}
```

Bjarne Stroustrup, Designer of C++

- Numerous awards, engineering and science
  - ACM Grace Hopper
- Formerly at Bell Labs
  - Now Texas A&M

- “There's an old story about the person who wished his computer was as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone.”

Unique words in C++

```c++
#include <iostream>
#include <fstream>
#include <set>
using namespace std;

int main(){
    ifstream input("/data/melville.txt");
    set<string> unique;
    string word;
    while (input >> word){
        unique.insert(word);
    }
    set<string>::iterator it = unique.begin();
    for(; it != unique.end(); it++){
        cout << *it;
    }
    return 0;
}
```

PHP, Rasmus Lerdorf and Others

- Rasmus Lerdorf
  - Qeqertarsuaq, Greenland
  - 1995 started PHP, now part of it
- Personal Home Page
  - No longer an acronym

- “When the world becomes standard, I will start caring about standards.”

Rasmus Lerdorf
Unique words in PHP

```php
<?php
$wholething = file_get_contents("file:///data/melville.txt");
$wholething = trim($wholething);
$array = preg_split("/\s+/", $wholething);
$uni = array_unique($array);
sort($uni);
foreach ($uni as $word)
    echo $word."<br>";
?>
```

Guido van Rossum

- BDFL for Python development
  - Benevolent Dictator For Life
  - Late 80's began development
- Python is multi-paradigm
  - OO, Functional, Structured, ...
- We're looking forward to a future where every computer user will be able to "open the hood" of their computer and make improvements to the applications inside. We believe that this will eventually change the nature of software and software development tools fundamentally.
  
  Guido van Rossum, 1999!

Unique Words in Python

```python
#!/usr/bin/env python
def main():
    f = open('/data/melville.txt', 'r')
    words = f.read().strip().split()
    allWords = set()
    for w in words:
        allWords.add(w)
    for word in sorted(allWords):
        print word
if __name__ == "__main__":
    main()
```

Kernighan and Ritchie

- First C book, 1978
- First "hello world"
- Ritchie: Unix too!
  - Turing award 1983
- Kernighan: tools
  - Strunk and White
- Everyone knows that debugging is twice as hard as writing a program in the first place. So if you are as clever as you can be when you write it, how will you ever debug it?

  Brian Kernighan
How do we read a file in C?

```c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int strcompare(const void * a, const void * b){
    char ** stra = (char **) a;
    char ** strb = (char **) b;
    return strcmp(*stra, *strb);
}

int main(){
    FILE * file = fopen("/data/melville.txt", "r");
    char buf[1024];
    char ** words = (char **) malloc(5000*sizeof(char **));
    int count = 0;
    int k;
    while (fscanf(file, "%s", buf) != EOF){
        int found = 0; // look for word just read
        for(k=0; k < count; k++){
            if (strcmp(buf, words[k]) == 0){
                found = 1;
                break;
            }
        }
        if (!found){ // not found, add to list
            words[count] = (char *) malloc(strlen(buf)+1);
            strcpy(words[count], buf);
            count++;
        }
    }
    qsort(words, count, sizeof(char *), strcompare);
    for(k=0; k < count; k++) {
        printf("%s\n", words[k]);
    }
    for(k=0; k < count; k++){
        free(words[k]);
    }
    free(words);
}
```

Storing words read when reading in C

```c
while (fscanf(file, "%s", buf) != EOF){
    int found = 0; // look for word just read
    for(k=0; k < count; k++){
        if (strcmp(buf, words[k]) == 0){
            found = 1;
            break;
        }
    }
    if (!found){ // not found, add to list
        words[count] = (char *) malloc(strlen(buf)+1);
        strcpy(words[count], buf);
        count++;
    }
}
```

Courses of Study

- What do you take first?
- What’s next?
- In between?
- Who teaches what?