Warm-up question

- In your own words, **what is research?**
"What is Research?"

Seminar Talk
In your own words, **what is research?**

(answers courtesy of my CS 470 class)
What is research?

- “Research is the process of finding information.”
- “Research is when you use your own time to understand and describe in words a topic you did not know about before.”
- “Looking for credible information pertaining to a specific topic.”
- “Utilization of academic, peer-reviewed publications in order to better understand or solve a problem.”
- “It's taking concepts or ideas and collecting valuable information pertaining to it, with some fact checking of course!”
What is research?

- “Working at the edge of knowledge in a field attempting to push that frontier a little further with your work.”
- “Thorough investigation into a subject, with the end result of finding new information.”
- “Research is building on the work of others on a topic of the researchers choice to posit new arguments and find new discoveries that might interest yourself or the general public.”
- “Learning new things then doing those things then writing about those things.”
What is research?

- The former is *secondary* research
  - Wikipedia: "summary, collation and/or synthesis of existing research"

- The latter is *primary* research
  - OECD 2015: "creative and systematic work undertaken to increase the stock of [human] knowledge"
  - Goal: **novelty**!
  - Many subcategories:
    - Purpose: *theoretical* vs. *applied*
    - Target: *formal* vs. *natural* vs. *social*
    - Methodology: *scientific* vs. *historical* vs. *artistic*
Knowledge (visualized)

all current human CS knowledge

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
all current human CS knowledge

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

M.S.

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

specializing

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

publish a paper!

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

world expert in topic XYZ!!

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
all current human CS knowledge

"wait, you're still in school?!?"

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

Ph.D. (yay!)

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Knowledge (visualized)

all current human CS knowledge

(the big picture)

based on http://www.happyschools.com/bachelors-vs-masters-vs-phd/
Another perspective

- As faculty advisors...
  - Undergrad projects: we have a reference solution
  - Graduate projects: we know a solution is possible
  - Research projects: we think a solution *might* be possible
• “Research is the process of systematically casting a fishing rod into the unknown and hoping that you reel in something worthwhile.

Sometimes you catch nothing, sometimes you get something worthwhile, and sometimes you get something that looks worthless until it's published by somebody else three years later.

But regardless you slowly begin to learn about the world on the other end of that hook.”
If that is what, now how?

- My advisor from UVA (Bill Wulf)
  - “We don’t really know how so we use the apprentice model.”
- More detailed references
  - How to pick an advisor
    - “Getting Started in Undergraduate Research”
  - How to read a paper
    - “How to Read an Engineering Research Paper”
- What I wish I knew/organization
  - “Organizing your Research and Developing your Research Skills”
  - “Everything I Wanted to Know about CS Graduate School at the Beginning but Didn’t Learn Until Later”
Research Interests
- Education (POGIL Project, Open Source)
- Workload Characterization
- Computer Architecture (E-Flynn)
Computer Architecture

1. Running requires building new idea.
2. Simulation necessary…but huge.7

Current Processor Design

118 Program/Workload

+ New Idea

118 Program/Workload

COMPARE

Measurements 47

Measurements 47
E-Flynn

### Flynn’s Taxonomy

<table>
<thead>
<tr>
<th></th>
<th>Single Data Stream</th>
<th>Multiple Data Streams</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Instruction</td>
<td>SISD</td>
<td>SIMD</td>
</tr>
<tr>
<td>Multiple Instruction</td>
<td>MISD</td>
<td>MIMD</td>
</tr>
</tbody>
</table>

```c
int binary_search( int nums[], int first, int last, int x) {
    while (first <= last) {
        int center = (first + last) / 2;
        if ( x > nums[center]) {
            first = center + 1;
        } else if (x < nums[center]) {
            Last = center -1;
        } else {
            return center;
        }
    return -1;
}

int main() {
    int nums[] = { 1, 4, 8, 14, 17, 19, 25, 27, 29, 48, 49, 51, 53, 57, 79};
    binary_search(nums, 0, 15, 14);
}
```

**Figure 2: C++ code for a binary search**

### Expanded Flynn’s Taxonomy

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>$2^2$</th>
<th>$2^3$</th>
<th>$2^4$</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic Block</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loop/Function</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Thread</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
My research interests (Mike Lam)

- Program analysis (CS 430, 432)
- Systems-level software tools (CS 261)
- High-performance computing (CS 470)
- Real number representations (CS 261)

I write software tools (primarily instrumentation-based) to analyze computer programs, with the goal of improving performance or improving accuracy.

Current projects:

- **CRAFT** ([https://github.com/crafthpc/craft](https://github.com/crafthpc/craft))
  Reports precision requirements for each instruction in a program
- **SHVAL** ([https://github.com/lam2mo/shval](https://github.com/lam2mo/shval))
  Simulates running programs with an alternate precision (e.g., double vs. single)
- **Automatic differentiation** (collaboration with LLNL)
  Rigorously quantify each input's effect on each output
- **Workload characterization** (brand-new collaboration with Dr. Weikle)
  Use instrumentation-based analysis to describe and categorize programs
Michael Stewart

Teaches:
• Intro (CS 149 and [eventually] 159)
• Interaction Design
• Web Development (eventually)
• Mobile App Development (eventually)

http://HCIentist.com
Human-Computer Interaction

Interdisciplinary research field centered around the way people use technology (or don’t) and the effects of technology use on people.

1. Gather qualitative and quantitative data about people and systems (evaluate)
2. Produce Implications for Design
3. Design
   1. Evaluate (gather qualitative and quantitative data about people and systems)
What new and meaningful ways can technology support us in connecting with others?

How can we utilize technology to design how people interact in different contexts (e.g. education)?

What experiences can we design with interfaces that we do not intend the user to interact with?
Process Oriented Guided Inquiry Learning in Introductory Computer Science

Helen Hu, Westminster College
Clif Kussmaul, Muhlenberg College
Chris Mayfield, James Madison University
Aman Yadav, Michigan State University
What is POGIL?

Process Oriented
(cooperative learning): conscious commitment to development of important process skills

Guided Inquiry
(constructivism): learning cycle activities

Model 1  Character Arrays

The primitive type `char` is used to store a single character, which can be a letter, a number, or a symbol. In contrast, the reference type `String` encapsulates an array of characters.

```java
char letter;
letter = 'A';

char[] array;
array = new char[]
{ 'c', 'a', 't' };

String word;
word = "dog";
```

Questions (15 min)  

1. How is the syntax of character literals and string literals different?

2. What is the index of 'd' in the string above? What is the index of 'g'? In general, what is the index of the last character of a string?
Related Work (in Chem Ed)

Grade Distributions in General Chemistry

Data (n = 905) from small (~24 students) sections of three instructors using lecture approach (1990-94) prior to implementation of POGIL pedagogy (1994-98).

Performance on Organic Chemistry 2 Unannounced First Day Pre-Quiz

All students passed Organic Chemistry 1 at this institution during the previous semester

All sections of Organic Chemistry 1 had more than 150 students.


5-Year Project in CS

I: Activities
- IntroCS POGIL activities
  - enhanced activities
  - # activities enhanced & used
  - better activities & learning

II: Prof Dev & Support
- POGIL workshops; faculty contexts
  - prof dev & support (teams, community)
  - # faculty trained & adopting
  - better prof dev, adoption & learning

III: Faculty Data
- knowledge of: learning strategies, adoption, communities, inclusion, etc
  - surveys/interviews observations
  - # surveys, etc qualitative data
  - better knowledge & understanding of strategies, prof dev, adoption, etc.

IV: Student Data
- surveys/interviews observations
- # students learning outcomes

Expected Outcomes:
- better faculty outcomes (motivation, use of strategies, effectiveness)
- better student outcomes (motivation, learning, retention, diversity)

Existing Factors:
- IntroCS POGIL Project

Long Term Impact:
- easier for more varied faculty to adopt POGIL & other strategies

Fig. 1: Logic Model
**Regular Solids (Platonic Solids)**
Dated to the Greeks
(The culmination of Euclid’s Elements)

All faces the same regular polygons.
All vertices indistinguishable.

**Semi-Regular Solids**
(Archimedean Solids)

All faces are regular polygons.
All vertices indistinguishable.

**Middle ages** — not so interested
**Moving into the 19th century** — mathematicians become interested again.
Specifically, are any flexible?
Cauchy (1813)
Convex polyhedra are globally rigid.

Bricard (1897)
Flexible, but self-intersecting, polyhedra exist.

Maybe all flexible polyhedra are self-intersecting?

Alexandrov (1958)
Extends Cauchy in some interesting and important ways.

Bellows Conjecture (unclear)
Any flexible polyhedron would not change volume during a flex.

Gluck (1974)
“Almost all” polyhedra are rigid.

Its not looking good for flexible, non-self-intersecting polyhedra.
SUPRISE!
Connelly (1977)
There exist flexible polyhedra!

Late 20th into 21st century
Explosion of interest in polyhedra and related flexible structures, especially in different geometries (particularly in hyperbolic geometry).

https://en.wikipedia.org/wiki/Steffen%27s_polyhedron
https://www.math.iupui.edu/~rroeder/HYPER-HEDRON/
http://www.geometrie.tuwien.ac.at/stachel/Albalulia_dr.pdf
MY RESEARCH
Flexibility of circle polyhedra — Tightly connected to flexibility of hyperbolic polyhedra

Collaborators: Philip L. Bowers (FSU), Kevin Pratt (Carnegie Melon)

UNDERGRADUATE RESEARCH PROJECT
Following Bricard’ classification of all flexible octahedra, can we derive a similar classification of all flexible circle-octahedra?

Collaborators on Koebe-Lib: Maddie Brower (JMU), Sarah Ciresi (Georgetown)

arsgeometricalab.com  github.com/johncbowers
3D Printed Spatial Graphs
Experimental mathematics designed with code

Dr. Laura Taalman
ARS GEOMETRICA

John Bowers's and Laura Taalman's Research Lab in Discrete Geometry
Classes, workshops, and programs run from JMU 3SPACE since we opened in Fall 2013, with current courses at the top. Programs with detailed websites are marked with asterisks. Click through and check out what we’ve been doing in class!
$fn = 30;

// Draw a complete bipartite graph with even k where k >= 2.
complete_bipartite (4, 25, \sqrt{2} * 25, thickness=2.5);

function rotate_about_z (p1, a) =
    [(p1[0]*cos(a)) - (p1[1]*sin(a)), (p1[0]*sin(a)) + (p1[1]*cos(a)), p1[2]];

module rod (point1, point2, thickness) {
    diff_vector = [point2[0] - point1[0], point2[1] - point1[1], point2[2] - point1[2]];
    radius = \sqrt{((diff_vector[0]*diff_vector[0]) + (diff_vector[1]*diff_vector[1]) + (diff_vector[2]*diff_vector[2]))};
    theta = \text{atan}((diff_vector[1]/diff_vector[0]));
    phi = \text{acos}((diff_vector[2]/radius));
    if (diff_vector[0] < 0) {
        rotate(point1, rotate(theta - 90, [0,0,1]) rotate(phi, [1,0,0])) cylinder(h = radius, r = \text{thickness}/2);
    } else if (diff_vector[0] > 0) {
        rotate(point1, rotate(theta + 90, [0,0,1]) rotate(phi, [1,0,0])) cylinder(h = radius, r = \text{thickness}/2);
    } else {
        rotate(point1, rotate(180, [0, 0, 1]) rotate(phi, [1, 0, 0])) cylinder(h = radius, r = \text{thickness}/2);
    }
}

module extra_edges (k, point1, point2, angle, thickness) {
    for (i = 0 : floor((k/2) - 1)) {
        rod(point1, rotate_about_z(point2, 2*i*angle), thickness);
    }
}

module complete_bipartite (k, radius, height, thickness) {
    // Set any constants used throughout but not passed as arguments
    vertex_thickness = 2*thickness;
    angle = 360/k;
}
// MODULES FOR GRAPHS
// Draws a general Petersen Graph where k | n
module even_p(n, k, outer_radius, inner_radius, thickness) {
    // Set any constants used throughout but not passed as arguments
    height_scale = outer_radius/k;
    vertex_thickness = 2*thickness; // was 1.5*thickness

    for(i = [0 : n]) {
        // Defines a set of vertices used to create the edges of the graph
        point = [outer_radius, 0, 0];
        rot_point = [outer_radius*cos((360/n)), outer_radius*sin((360/n)), 0];

        // CHOOSE ONE OF THE BELOW
        // Changes the parity of the height depending on whether or not the cycle's mod k number
        // is less than half the number of cycles. Results in up to half of the cycles being drawn below
        // the base
        parity = 1;
        // parity = (((i/k) < (k/2)) ? 1 : -1;
        point_prime = [inner_radius, 0, parity*(parity*height_scale + (((i/k)*height_scale))];
        rot_point_prime = [inner_radius*cos(k*(360/n)),
                           inner_radius*sin(k*(360/n)),
                           parity*(parity*height_scale + (((i/k)*height_scale))];

        translate(l*(360/n), [0, 0, 1]) {
            union() {
                // Draws the edges of the graph using the rod function
                // color("blue")
                rod(point, rot_point, thickness);
                // color("green")
                rod(point, point_prime, thickness);
                // color("red")
                rod(rot_point_prime, rot_point_prime, thickness);

                // Draws the vertices of the graph using translated spheres that are slightly
                // thicker than the thickness of the graph's edges
                translate(point) sphere(vertex_thickness/2);
                translate(point_prime) sphere(vertex_thickness/2);
            }
        }
    }
}

Viewpoint: translate = [-0.97, 3.76, 5.46], rotate = [55.00, 0.00, 25.00], distance = 213.38
JMU CS Undergraduate Research Page

goo.gl/AVQ95g

(will be updated later today)