Introduction to Computer Science

Dr. Andy Gill

Electrical Engineering and Computer Science

December 1st and 5th

Computer Science

In this class, we are going to

- **Learn** what it means to program a computer
- **Learn** basic programming
- **Enhance** an existing Program (a simple game)
- **Program** a bot (give a virtual agent “smarts”)

Friday is a lab, in the engineering commons.

First - who is Andy Gill?

- Researcher in the field of programming languages
- Ph.D. from the University of Glasgow
- Spent 12 years in industry as a compiler engineer and technical lead
    - Worked on a Java compiler, a C++ compiler, and a Haskell compiler
    - Co-founded a technology startup in Portland, OR, that used Haskell
- Teaching since 2002; moved to KU in 2008.
- Want to share my research interests: Computers and Programming Languages

What is a Computer?

“The trouble with computers is, they’re very sophisticated idiots. They do exactly what you tell them at amazing speeds.”

— The 4th Doctor.

Why do we have computers?

?

Computer Science is **not**

Computer Science is **not** about programming.

- CS uses programming as a tool to get a job done.
- CS uses programming in the same way most engineering subjects use Math.
- Furthermore, programming is its own type of Math.
- Programming can be executable and/or declarative.

We are going to see some basic programming today.

What does Computer Science using **programming** for?

- Mechanization — Example: HVAC controller
- Simulation — CS helping others
- Graphics — User Interfaces
- Search — Finding things
- Storage — "The Cloud"
- The Web — Online shopping carts

All use programming as a means to an end.

What does Computer Science using **programming** for?

Think of a single digit number.

1. Square it.
2. Add the result to your original number.
3. Divide by your original number.
4. Add, oh, how about 17.
5. Subtract your original number.
6. Divide by 6.

We are going to automate (program) this.

Programming “Think of a number”

```
// Think of a number.
x = 9;
// Square it.
a = x * x;
// Add the result to your original number.
a = a + x;
// Divide by your original number.
a = a / x;
// Add, oh, how about 17.
a = a + 17;
// Subtract your original number.
a = a - x;
// Divide by 6.
a = a / 6;
// Print the number to the screen.
console.log(a);
```

2nd Problem - find your direction

- You are on a 2-dimensional plane.
- You want to point towards another (given) point on the plane.
- How do you compute this?

if x, y are your coordinates, then $atan2(y, x)$ is the direction you are from the origin, in radians.

Programming “Find your direction”

```
// Original Coords.
x = 3; y = 4;

// Target Coords.
xt = 5; yt = 2;

// Take the difference between the vectors.
xd = xt - x;
yd = yt - y;

// Compute the radian
r = atan2(yd,xd);

// Print the result.
console.log(r);
```

Types of expressions

Expressions are on the right hand side of assignments.

Numbers:

- **1, 2, -1, -2, . . .**

Arithmetic:

- +, −, ∗, /

Parenthesis:

- (. . .)

Assignment Statements

$$\text{var} = \text{expr} \;;$$

var is a variable name. letters and numbers, starting with a letter. Upper case is for constant variables, like **PI**.

expr is an expression, like $\mathbf{x + 1}$. Complex expressions are allowed, as are parenthesis. expressions can contain calls to functions, like **atan2**.

Remember to close with semi-colon.

Controlling the boat

Task:

- Make the boat go North.
- (Get eaten by monster.)

- Question: Is there a way of escaping?

What if we want to do something conditionally?

Conditionals are a way of making choices.

Task:

- Assign **x** to -1,0 or 1, depending if **y** is negative, zero, or positive.
- Split this problem into three parts first.

Computing absolutes

```
y = 0;
if (y < 0) {
    x = -1;
}

if (y == 0) {
    x = 0;
}

if (y > 0) {
    x = 1;
}
console.log(x);
```

Form of a conditional

```
if (condition) {
    assignment;
    assignment;
    assignment;
}
```

Types of conditions

The boring ones:

- **true, false**

Equality and inequality:

- `==`, `!=`

Comparisons:

- `<`, `<=`, `>`, `>=`

not!

- `!`

Chaining comparisons:

- `&&`, `||`

Parenthesis:

- (. . .)

Aside. . .

- Almost everything in computer science can be nested.
- **if** statements can contain **if** statements.
- **if** statements can contain **if** statements that contain **if** statements.
- . . .

New task

Control the boat.

- Make the boat go North.
- After 100 *steps*, make the boat go West.

Advanced:

- Make the boat go North.
- After 100 *steps*, make the boat turn West **gradually**.

Assignment Statements and Conditionals

- We have learned about **assignment statements**.
- Assignments give a new value to a named variable.
- Assignments compute this new value.

- We have learned about **conditionals**.
- Conditionals *optionally* do assignments.

- On Monday
  - Lab (in Eaton 1014 / 1018)
  - Escape from Crater Lake!