Data Structures with C++ : CS189

Lecture 1-1: Inheritance Review

Welcome!

- This class is hard
- There will be parts that aren't in the book or on the web because they require your creativity and dedication
  - Which is the whole point of a programming job
- But first, the administrative stuff
  - Enrollment
  - Logins
  - Canvas Tour
    - Stressing the Announcements
  - Syllabus reading
  - Syllabus Quiz
    - Anyone who doesn't take it gets dropped

Guidelines for Success

- How to get a good grade
  - Don't ever quit
  - Don't leave early
  - Do ask questions
- You aren't supposed to know any of this
  - The point of school is to make you know it
  - The point of me is to help you do that
- The average grade in this class will shock you
  - It's an A. People either drop or understand it. I am the perfect filter class.

Important Life Acronyms

- DSPS
  - For requesting accommodations for learning difficulties
- EOPS
  - For help with life/money problems - not financial aid
- SEP
  - Counseling helps make sure you are taking the right classes

Your Professor

- Professional programmer for fifteen years
    - This class is more programs than tests
- Teaching style follows the brain pattern theory
    - Eyes: Pictures on board
    - Ears: Listen to lectures
    - Hands: Give examples
    - You can find as many papers for this as against this. I've found it works
- If you feel I am not reaching you, you need to tell me

Inheritance

Depending on your previous class, this is either super new or super review

Object Review

- Classes are templates to make Objects
- Objects have Properties to define what they are, and Methods to define what they can do
- C++ is all about Object Oriented Design, so they are incredibly important
- The next two weeks are all about making them even more powerful.

Cohesion

```
class TV
{
Remote *X; //Bad.
int size;
string brand;
TVShow *Y; //Bad
```

Everything TV is here
Nothing not-TV is

- Object Modeling is the art of turning a problem in to classes you can implement to solve it
- Cohesion is the idea that a class should know everything about itself and nothing else
- Bad cohesion means that if you change class A, you have to go fix B and C

Information Hiding

```
class TV
{
	int watts;
	bool isOn;
	float capacitance;
	int refreshRate;
public:
	void FlipPower();
	void SetChannel();
};
```

- How can objects interact if they aren't allowed to know each other works?
  - A Remote knows how to turn on a TV
  - A TV knows how to be turned on
- A Remote doesn't need to know what electricity is (cohesion) and it doesn't know how to actually make the TV put pixels on the screen
- A class hides everything other classes don't need to know about it

Simple Objects

```
void Cookie::Cook()
{
if( mFlavor == "X")
...
else if( mFlavor == "Y")
...
else if( mFlavor == "Z")
...
else if( mFlavor == "A")
...
else if( mFlavor == "B")
...
```

- I have a Cookie class
- Later I need ten types, and one of the ten cooks faster than the others
- Then I have twenty types, and three of them are smaller
- At the end of the year, so many special cases have been added that adding new types or debugging anything is difficult

ISA

```
class OatmealCookie
	: public Cookie
{
public:
	void Cook();
};

class Cookie
{
public:
	void Eat();
};
```

- ISA is a made up word, but it perfectly describes the whole week
- Cookie class is too big? Make an OatmealCookie class that ISA Cookie
- Everything Oatmeal-specific goes in OatmealCookie
  - Cohesion!
- OatmealCookie doesn't know how any other type of cookie works
  - Information Hiding!

Which Gets Called

```
OatmealCookie X;
X.Cook(); // Oatmeal
X.Eat();    // Regular
```

- If you try to call Eat on an OatmealCookie, the code tries there first but can't find it
- Then it thinks "Hey, an OatmealCookie ISA Cookie. I'll look in Cookie."
- If you call Cook with an OatmealCookie it finds it like normal
- There are a few more complicated ways to connect these classes, but that's next week

Smaller Details

- You can't inherit from more than one class, and you don't know about your grandparents
- Private is the same. Only someone of your class can use private stuff
  - Information Hiding!
- "Protected" means that any subclasses can go ahead and use it, but honestly I've never seen this used.
  - Breaks cohesion

Go to Canvas Activity

Get used to this system. After every topic I throw up a small multiple choice quiz to see where everyone is. If everyone aces it, I can move on. If everybody bombs, I'll go back.

End

First class is usually eaten by admin stuff so I stop here