

Dynamic Dispatch, part 5

The last step: what if we had a class where we didn't WANT to ever implemented it?

Why would we do such a thing?

- Class relationships are a POWERFUL tool.
- You can have a base class pointer point to anything derived from your base class! This is useful.

Lets make class brian an abstract class. Here's how:

```
// class spec for brian class  
  
class brian {  
public:  
    brian( ) { cout << "I know C++. . ." << endl;}  
    virtual void speak( ) = 0;  
};
```

Now, we can no longer instantiate an instance of class brian. In other words, if the client writes:

```
brian BRIAN;  
or  
brian * BRIAN = new brian();
```

These will cause a compile error.

This now forces any class derived from brian to implement the speak() method. You can make entire classes that contain nothing but abstract virtual functions, and then use instances of that class to point to anything derived by that class.

This is INCREDIBLY useful! More on this soon. . .