

Inheritance and Polyorphism

Joel Willoughby

COP 3503

Outline

1 Introduction

2 This Lab

3 Wrapping Up

Agenda

- Inheritance Ideas
- Inheritance in C++
- Polymorphism

Outline

1 Introduction

2 This Lab

3 Wrapping Up

Inheritance

- Often times in programming, we want to add functionality to an existing class while keeping that classes previous behavior
- This is usually due to having some base class that can be extended in different ways
- Inheritance is a good way to go about this

C++ Syntax

```
1  class Base{
2  protected:
3      int m1;
4  public:
5      virtual int fun1();
6      void fun2();
7  };
8
9  class Derived: public Base{
10     // Now has access to all the
11     // protected members in class Base
12  public:
13      virtual int fun1();
14      void fun2();
15  };
```

Protected

- The protected keyword is a modifier like public or private
- It keeps the things private from all code except for the class it is in as well as any classes that inherit from it
- This is perfect for things in the Base class that you want access to in the Derived class

Virtual

- The virtual keyword basically tells the compiler that inheriting classes will probably need to override this method
- You can still put valid code in a virtual function
- The virtual function in a parent class is basically a default to use in case there is nothing better. By better, we mean more derived or specialized.
- This allows for polymorphism

Outline

- 1 Introduction
- 2 This Lab
- 3 Wrapping Up**

Questions

???