

## **Inheritance**

**2-11-2002**

---

---

---

---

---

---

---

---

### **Opening Discussion**

- What did we talk about last class? Did you happen to look at the description of assignment #2.
- What is inheritance? What does it do for us? What does it signify in our programs?

---

---

---

---

---

---

---

---

### **What is Inheritance?**

- This is a very hard question to answer directly. The best way I know to do it is to explain what it does.
- Part of the difficulty comes from the fact that inheritance actually does two distinct things. In C++ these roles are not well separated and are unnecessarily complicated. This is in part because the two roles hadn't been well distinguished when C++ was created.

---

---

---

---

---

---

---

---

## **Inheritance as Subtyping**

- The primary use of inheritance in programming languages is to denote subtyping relationships.
- You use subtyping all the time in your life you just don't think about it. In programming we have to be more formal.
- Type B is a subtype of A if any time we expect something of type A an object of type B can be used.

---

---

---

---

---

---

---

---

## **Is-A vs. Has-A**

- Subtyping and inheritance denote an Is-A relationship, not a Has-A relationship.
- A Car is a Vehicle and a Toyota is a Car so these are valid subtypes and those classes could inherit from one another.
- A Car Has-A BrakePeddle, but it isn't one. This type of relationship is generally signified by putting a BrakePeddle member in Car.

---

---

---

---

---

---

---

---

## **Inheritance for Code Reuse**

- Inheritance also allows us to get extra code reuse with out classes for the code in the methods of the classes.
- By default, a method in a superclass is also a method of the subclass and can be called for objects of the subclass's type.
- We will talk more about the details of this next class.

---

---

---

---

---

---

---

---

## Inclusion Polymorphism

- Inheritance also gives us another form of Universal polymorphism called inclusion polymorphism. This springs directly from the definition of a subtype given previously. If a method in any class is written to accept an object of the base class, then it can also accept any object of any subclass of that base class.
- While this can be slightly more limiting than templates because it restricts the types that can be used, it also allows dynamic binding which templates don't. More on this next class.

---

---

---

---

---

---

---

---

## Syntax of Inheritance in C++

- Denoting inheritance in C++ is quite simple. In the declaration of the subclass you simply follow the name of the class by a colon, a visibility specifier, and the name of the base class it inherits from.
- We will typically be using public as the visibility for inheritance. Without it you kill some of the inherent polymorphism.

---

---

---

---

---

---

---

---

## Minute Essay

- In what ways might inheritance be used in the project code? I intend for you to use quite a bit of inheritance in your code. You can look at the assignment #2 description to see an answer of this later on.
- Inheritance is quite tricky to fully understand in C++. I strongly suggest you do the reading for this chapter.

---

---

---

---

---

---

---

---