University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 220: Computer Systems & Programming

Data and Function Inheritance

Can Organizing Data & Specializing Functions be Easier?

The **organization of data** and **function specialization** that you have just seen
- **helps when developing large systems in C**,
- but **puts the burdens** of learning and following best practices **on the programmer**.

In about the 1980s,
- people started thinking about **automating these tasks**
- to **enforce best practices** and
- to **reduce oportunities for** human **error**.*

*I saw it, but it was too ironic to fix.

C++ Uses Data and Function Inheritance to Automate

In particular, why not have **the programmer**
- **specify only** the **type hierarchy and**
- **which functions should be changed** for a subtype?

The compiler can the lay out the data, create the virtual function tables, and so forth.

C++ performs such automation,
- leveraging **data and function inheritance**
- to produce structures and functions that
- usually look exactly the same as one would produce in **C**.

Parent is the Base, Child is the Derived Type/Class

What are data and function inheritance?

Think back to our **type hierarchy**.

In any given parent-child relationship,
- the **parent** is called the **base type or class** (or, historically, the super-type or class), and
- the **child** is the **derived type or class** (or, historically, the sub-type or class).

