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

ECE 220: Computer Systems & Programming

Implications of Overloading

Overloading Implies Rules for Matching Calls to Functions

Since functions can be overloaded,
- a **compiler must have rules**
- **for choosing** amongst functions
- with the same name.

Let's first ask: **how different**
- **do** two **definitions** of a function **need to be**
- for the compiler **to distinguish them?**

C++ Supports Extremely Minor Variations

High-level answer:
- **C++ allows for extremely minor variations**;
- use them at your own risk.

Variations on functions
- are not much different
- than naming two variables
- VaRiAbLe and vArIaBlE.

C++ Distinguishes Between Similar Types

For example,
- C's **default conversions** are **not assumed**,
- so the following can be differentiated:

```
int operator+= (int i);
int operator+= (char c);
```

One can also distinguish
- between signed and unsigned values,
- between `const` and non-`const` values,
- and so forth.

