CSE 333
Lecture 10 - cleaning up some details

Steve Gribble
Department of Computer Science & Engineering
University of Washington
struct vs. class

in C

- a struct contains only fields
  ‣ cannot contain methods
  ‣ does not have public vs. private vs. protected

in C++

- struct and class are (nearly) the same
  ‣ both can contain methods
  ‣ both can have public vs. private vs. protected

- **struct**: default public,  **class**: default private
When to use references?

A stylistic choice

- not something mandated by language

Google C++ style guide suggests:

- input parameters:
  ‣ either use values (for primitive types like int)
  ‣ or use const references (for complex structs / object instances)

- output parameters
  ‣ use const pointers
virality of const

- **OK to pass**
  - a pointer to non-const
- to a function that expects
  - a pointer to const

- **not OK to pass**
  - a pointer to a const
- to a function that expects
  - a pointer to a non-const

```cpp
#include <iostream>

void foo(const int *y) {
    std::cout << *y << std::endl;
}

void bar(int *y) {
    std::cout << *y << std::endl;
}

int main(int argc, char **argv) {
    const int a = 10;
    int b = 20;

    bar(&b);   // OK
    bar(&a);   // not OK

    return 0;
}
```