References in *C*++

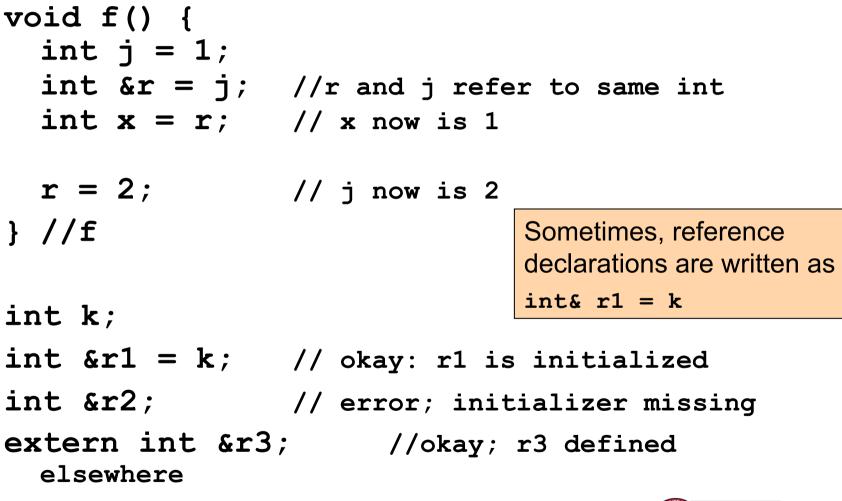
• **Definition** *Reference:— An Alternative Name for an Object*

BIG difference from Java

- *References* are only created in declarations and parameters
- A reference can only appear where the object itself could have appeared

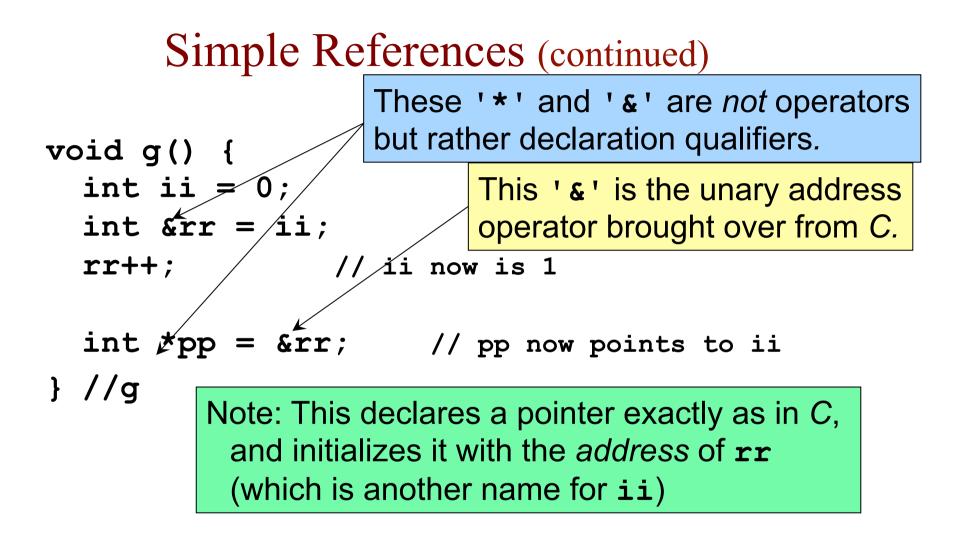


Simple References



CS-2303, C-Term 2010







Reference Parameters

- An *alias* for its corresponding argument in a function call.
 - & placed after the parameter type in the function prototype and function header
- Example
 - int &count in a function header
 - Pronounced as "count is a reference to an int"
- Parameter name in the called function body actually refers to the original variable in the calling function.



Reference Parameter Example

C version

 void swap (int *a, int *b) {
 int temp = *a;
 *a = *b;
 *b = temp;
 } // void swap(...)

Hazard: a **NULL** pointer

• C++ version

void swap (int &a, int &b) {
 int temp = a;
 a = b;
 b = temp;
} // void swap(...)

Non-hazard: no pointer here



Notes on References and Pointers

- Pointers in *C* do multiple duty
 - *Links*, as in linked lists and trees
 - *Parameters*, where the function needs to return a value to an argument provided by the caller
 - Short-hand, a short way of referring to an object that otherwise would need a complex expression

- ...



Java vs. C++ References

- In Java, a reference is a data type.
 - It can be assigned to, compared, copied, stored, etc.
 - Same reference can refer to different objects at different times during execution
- In C++, a reference is an *alias* for an object
 - It cannot be assigned to; assignment is *through* the reference to the underlying object
 - Similar to dereferencing a pointer in C
 - A reference *always* refers to the same object for the duration of its scope



Repeat Three Times

A reference is not a *pointer*, ...

A reference is not a pointer, ...

A reference is not a pointer, ...

And neither of them resembles a Java reference

CS-2303, C-Term 2010



Questions?

CS-2303, C-Term 2010

