PIC 10B Discussion
Week 3

Professor: Michael Lindstrom
TA: Thomas Tu
R-, GL-, PR-, X-, and L- Values
R-, GL-, PR-, X-, and L- Values

• Two main types: R-values and GL-values
• R-values: “right hand side”
  • \( x = 3; \) // 3 is on the right hand side
  • \( y = x; \) // \( y \) is on the right hand side
• GL-values: “generalized left hand side values”
  • \( x = 3; \) // \( x \) is on the left hand side
  • \( \text{arr}[0] = 15; \) // \( \text{arr}[0] \) is on the left hand side
R-, GL-, PR-, X-, and L- Values

• R-values have two forms:
• PR-values: “pure R-values”
  • Literals: 5, “hello”, 0x4c392a
  • Return types that are not objects/references
    • vec.size(); // returns an int
• X-values: “eXpiring values” – stored temporarily
  • Return types that are references to R-values
    • string&& obj.getRvalueReference();
  • Non-reference R-value attributes
    • rvalue.attribute
  • Rvalue array subscripts rvalue[n]
R-, GL-, PR-, X-, and L- Values

• GL-values can be X-values or L-values
• L-values – “left values”
  • Functions or objects
  • References to L-values
  • Ball* ballptr;
  • *ballptr = otherBall;
• X-values
  • getFirstElement().m = 5;
&& References

• && references
  • Bind to r-values
  • string&& r_ref = “right”;

• Allows you to distinguish gl-values from r-values
  • Foo( Ball& gl_ball);
  • Foo( Ball&& r_ball);
  • So you can make certain operations more efficient
Why do we have all these types?

• Originally, just L-values and R-values, simple
• Wanted more efficiency, eliminate unnecessary copying
• Move semantics
• Now, we wanted some L-values to act like R-values
  • So we can reuse all the code for moving R-values
  • And thus, X-values were created
Move Semantics

• When you say “a=b”, what really happens?

• Does it “move” the object to you, or return a copy?
  • If move semantics are implemented, it can give you the original
  • This is good: for example, “a=arrayConcat(b,c);”

• When is it possible to move instead of copy?
  • If the object is an R-value
    • Remember that X-values are also R-values, as well as GL-values
  • If the object specifies the move functions
R-, GL-, PR-, X-, and L- Values

• How to tell the difference?
• Has identity: could you find whether two of these were stored in the same block of memory?
• Can be “moved” from – move semantics
• L-values: cannot be moved from, have identity
• X-values: can be moved from, have identity
• PR-values: can be moved from, do not have identity