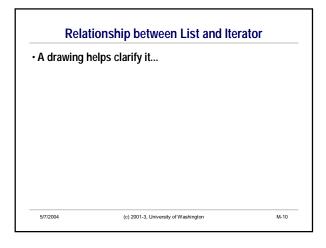


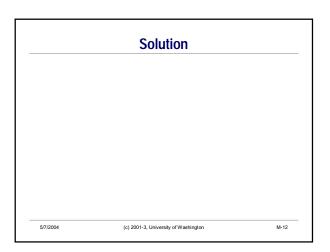
ArrayList names =;	
System.out.println("The names are	as follows:");
Iterator iter = names.iterator();	// get the iterator for the collection.
while (iter.hasNext()) { String name = (String) iter.nex System.out.println(name);	// while there is another element t(); // get the element (and cast it if needed // do something using the element.
}	// then go back to the top.
// Otherwise, we're done.	



Example: Finding the Longest Name Suppose we want to find the longest name in a list. How would we do it? Recall: "Bob".length() == 3 What's the algorithm in English? What's the Java code?

(c) 2001-3, University of Washington

M-11



5/7/2004



 We can also process an ArrayList using get(index) for (int k = 0; k < names.size(); k++) { process names.get(k);

} • Tradeoffs

5/7/2004

 Iterators are more general – work on all collections, even if the collection doesn't support indexed access (i.e., using get(k) to access elements directly)

 Iterators by default only support traversal of a collection from beginning to end.

Some types of collections have special iterators to allow going in reverse

General rule: use iterators (the more general solution) normally;
 use other traversals when iterators don't do what you need

(c) 2001-3, University of Washington

M-13