For Loops
AP Computer Science

Traditional For Loops Used to Access Every Item in an ArrayList

//This assumes that you have not declared your ArrayList to hold just one type of object

ArrayList myList;
myList = new ArrayList();

//The loop accesses every item in the ArrayList based on its position in the list
//Note that you must typecast when you retrieve the item from the list

for (int x = 0; x < myList.size(); x++)
{
    AquaFish bob = (AquaFish) myList.get(x);
    //more code here to work with bob
}

Enhanced For Loops Used to Access Every Item in an ArrayList

//This assumes that you have declared your ArrayList to hold only one type of object

ArrayList<AquaFish> myList;
myList = new ArrayList<AquaFish>();

//The loop does not access items based on position...this is more like an iterator
//No typecasting is required; all items are already guaranteed to be the same type

for (AquaFish bob: myList)
{
    //more code here to work with bob
}

The generic format for an enhanced for loop is as follows:

    for (name_of_class    name_of_object    :    name_of_data_structure)
    {
                        
    }

Although the enhanced for loop can make code much clearer, it can't be used in some common situations. (from http://leepoint.net/notes-java/flow/loops/foreach.html)

- **Only access.** Elements cannot be assigned to, eg, not to increment each element in a collection.
- **Only single structure.** It's not possible to traverse two structures at once, eg, to compare two arrays.
- **Only single element.** Use only for single element access, eg, not to compare successive elements.
- **Only forward.** It's possible to iterate only forward by single steps.
- **At least Java 5.** Don’t use it if you need compatibility with versions before Java 5.