ArrayList Reference sheet

Wrapper Classes

int → Integer
double → Double
char → Char
boolean → Boolean

Setting up Your ArrayList

ArrayLists can only store objects; not primitive types (int, double, char, boolean.) Fortunately all of our primitive types have wrapper classes that you can use to easily convert to and from. So, if you want to have an ArrayList that stores a bunch of ints, use Integer. Setting up your ArrayList looks like:

```
ArrayList<Integer> nums = new ArrayList<Integer>();
ArrayList<Double> costs = new ArrayList<Double>();
```

Basic ArrayList Operations

```
size()                                  - number of elements
contains(Object element)                - returns true if element is in list
add(Object element)                     - adds the element to the list
set(int index, Object element)          - sets position index to element
get(int index)                          - returns element at position index
remove(Object element)                  - removes the element from the list
remove(int index)                       - removes the element at index position
indexOf(Object o)                       - returns the position of element o in the List
subList(int from, int to)               - returns a sub-list (like a substring)
Collections.sort(List)                  - Sorts elements in ascending order
```

ArrayList Example

```
import java.util.*;

public class Tmp {

    public static void main(String[] args) {
          ArrayList<Integer> numList = new ArrayList<Integer>();
          int []nums = {4,5,6,7,8,9};

          for(int i:nums){          // i BECOMES each element in nums
                numList.add(i);     // the ints are automatically converted to Integers
          }

          // If I just need to look at each element (like finding a sum) this
          // sort of for loop works fine.
          int sum=0;
          for(int i:numList){
                sum += i;
          }
          System.out.println("Sum = " + sum);

          // If I need to modify the numbers (e.g. add 10 to each) I'll need to
          // get the index of each element.
          for(int i=0; i<numList.size(); i++){
                int x = numList.get(i) + 10;
                numList.set(i, x);
          }
    }
}
```