

## ArrayLists Reference for Hangman

---

Based on a handout by Patrick Young

This handout gives you a quick reference for some of the concepts related to ArrayLists that may be useful to you for implementing Part III of the Hangman assignment.

### ArrayLists

You can think of an ArrayList as being a special object which is used to store a list of other objects. In your case, you'll be using an ArrayList to store the list of `strings` which can be chosen as words for Hangman.

### Using ArrayLists

ArrayLists are defined in the `java.util` package. To use ArrayLists, you'll need to import the package:

```
import java.util.*;
```

### Declaring an ArrayList variable

ArrayLists can store a variety of different types of information—for example, you can create an ArrayList of `GOvals`, an ArrayList of `GRects`, or in the case of Hangman an ArrayList of `strings`.

When declaring a variable of type ArrayList, you'll need to specify what type of data is stored in the ArrayList using a special angular bracket notation. Your declaration will look something like this:

```
private ArrayList<String> wordList;
```

The information in the angular brackets tells Java that this is an ArrayList of `strings`. ArrayLists themselves are objects and are created using constructors, just as `GOvals` and `GRects` are created using constructors. Here is how we would create a new ArrayList and assign it to our variable `wordList`:

```
wordList = new ArrayList<String>();
```

### Adding elements to your ArrayList

Call the `add` method to add elements to your ArrayList. Because your ArrayList has been declared to contain `strings`, you can call the `add` method with a `string` as an argument:

```
String word = readLine("?");    // reading a String from the user  
wordList.add(word);
```

### **Accessing elements of your ArrayList**

To access an element of your ArrayList, you can call the `get` method, passing in the index of the element you want to access. Remember, ArrayLists (as with regular arrays in Java) are zero-indexed—in other words, the first element in the list is actually at index 0, the second element is at index 1, and so forth. Because your list is declared to contain `Strings`, Java assumes that the element returned by `get` is a `String`:

```
int index = 0;    // index of first element in ArrayList
String word = wordList.get(index);
```

### **Determine the size of your ArrayList**

Your ArrayList will keep track of the number of element which you have added. To determine the number of elements in the ArrayList, call the `size` method:

```
int totalCount = wordList.size();
```

### **Chapter 11.8 is your friend for more information about ArrayLists**

There's a lot more to learn about ArrayLists. This handout is just providing a quick references for the Hangman assignment. We will cover ArrayLists more extensively in class and you can read more about ArrayLists in Chapter 11.8 of the class textbook.