**CIS 162 Lab 10**

**ArrayList of Strings and Dice**

**Objectives**

After completing this lab, you should be able to:

* *write* short problem solutions involving ArrayList

**Activity #1 Words**

Create a new class called Words that includes an ArrayList of Strings as an instance variable.

private ArrayList<String> info;

* public Words( ) - instantiates the ArrayList.

info = new ArrayList<String>();

* public void addWord(String str) - add the provided String to the end of the list.
* public void addWords(int num) - Prompt the user to repeatedly enter num words using a dialog . Add each word to the ArrayList in the order it has been provided.

**Using Dialog Boxes**

Instead of using the Scanner to read from the keyboard as we have done in the past, use dialog boxes.

Example:

String str = JOptionPane.showInputDialog("Enter a word: ");

* public void display() – print the list to the terminal window. This is one line of code.
* public boolean removeWord(int index) – remove the word at the requested index (return true). Do not attempt to remove a word if the index is not valid (return false).
* public int search(String str) – search the list for parameter str. Searches should work regardless of case. For example, “GVSU” is equivalent to “gvsu.” Hint: a String object has a method called equalsIgnoreCase(). Return the first index found or return -1 if not found.
* public void removeEveryOther() – Remove every other String from the ArrayList starting at index 0. This will have no effect on an empty list. A list of size one will return an empty list.

Before [apples, bananas, coffee, dressing, eggplant]

After [bananas, dressing]

**When working, copy to the appropriate Ch 10 zyLab.**

**Activity #2 Dice**

Create a new class, called Dice, that includes an ArrayList of GVdie as an instance variable.

private ArrayList<GVdie> allDice;

1. public Dice(int num) – Constructor instantiates an ArrayList and then instantiates each of the num dice. For example, this could create three dice or thirty.

allDice = new ArrayList<GVdie>();

public void roll() – Use a for-each loop to cycle through and roll each die.

1. public ArrayList<GVdie> getDice() – return the ArrayList in one line of code.
2. public int getTotal() – use a for-each loop to cycle through and add the sum of all dice.
3. public int getCount(int val) – return the number of dice that have the value of val. Return -1 if val is not a valid die value (1-6).

**When working, copy to the appropriate Ch 10 zyLab.**

**Activity #3 Integrate with existing GUI**

Now that the Dice class is working, add the provided DiceGUI.java to your project. It should compile and run with no changes.

**When working, demonstrate to your instructor.**

**Grading Criteria**

This lab is worth 10 points including Zylabs