

**CSCI 1101 – Winter 2017**  
**Lab. No. 6**

Notes:

1. All submissions must be made on Brightspace ([dal.ca/brightspace](http://dal.ca/brightspace)).
2. **Submission deadline is 11.55 p.m. (5 minutes to midnight) on Saturday, March 4, 2017.**
3. Put the java source code files and the text outputs for each exercise in a folder. Zip the folder into one file and submit the zip file.

**4. Marking Scheme:**

Each exercise carries 10 points. Your final score will be scaled down to a value out of 10.

Working code, Outputs included, Efficient, Comments included: 10/10

No comments: subtract one point

Unnecessarily long code and inefficient program, improper use of variables: subtract one point

No outputs: subtract two points

Code not working: subtract up to six points depending upon the extent to which the program is incorrect.

**5. Error checking:** Unless otherwise specified, you may assume that the user enters the correct data types and the correct number of input entries, that is, you need not check for errors on input.

**6. Testing your code and generating the outputs:** If the test data is included in the question, use that to test your classes. In addition, test it with two more input sets. Otherwise, create your own test data and run your program for at least 3 input sets such that they cover the range of results expected.

**Part 1: If you have not yet submitted the solutions to Lab 5, please include those in this week's submission. If you have already submitted Lab 5, no need to repeat the submission.**

**Part 2: ArrayLists**– Here's a simple test program that creates an ArrayList of String objects and manipulates them. Try it out. Make modifications to the method calls and understand their operations.

**No submission required for this exercise.**

**Warning: Cutting and pasting the code may cause syntax errors!**

```
import java.util.ArrayList;
public class ArrayListTest {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>(); // Create a list

        // Add elements to the list
        list.add("America"); // Add it to the list
        System.out.println("(1) " + list);

        list.add(0, "Canada"); // Add it to the beginning of the list
        System.out.println("(2) " + list);

        list.add("Russia"); // Add it to the end of the list
        System.out.println("(3) " + list);

        list.add("France"); // Add it to the end of the list
        System.out.println("(4) " + list);

        list.add(2, "Germany"); // Add it to the list at index 2
        System.out.println("(5) " + list);
    }
}
```

```

list.add(5, "Norway"); // Add it to the list at index 5
System.out.println("(6) " + list);

// Remove elements from the list
list.remove("Canada"); // Same as list.remove(0) in this case
System.out.println("(7) " + list);

list.remove(2); // Remove the element at index 2
System.out.println("(8) " + list);

list.remove(list.size() - 1); // Remove the last element
System.out.println("(9) " + list);

if (list.contains("France"))
    System.out.println("France found at index: " + list.indexOf("France"));

System.out.println("What is the element at index 1 ? : " + list.get(1));
}
}

```

**Exercise 1:** Write a program that reads words into two ArrayLists list1 and list2 and then creates a third ArrayList that contains words which are common to both list1 and list2. You may assume that the strings are entered as words separated by a space on a single line and the end is signaled by “-1” (String -1). You can use keyboard.next() to read each word.

A sample dialog is shown below:

```

Enter words on one line, end with -1
java c pascal ada java c++ -1
Enter words on one line, end with -1
c pascal java lisp lisp -1
[java, c, pascal, ada, java, c++]
[c, pascal, java, lisp, lisp]
Array List with common strings:
[c, pascal, java]

```

**Exercise 2:** Write a program that reads words into an ArrayList list1 and creates another ArrayList list2 that has the same words in list1 but no duplicates.

A sample dialog is shown below:

```

Enter words on one line, end with -1
java c pascal ada java java ada c++ -1

Array List with no duplicates:
[java, c, pascal, ada, c++]

```