# Central Washington University
# College of the Sciences
# Department of Computer Science

CS-301 Data Structures Fall 2016

Lab Practice 03

*We will do the basic practice of the JFC. Also we will practice the creation of inner classes implementing* `Iterator<E>`.

```
lb03.pdf
DataStruct.java
TriangleDS.java
Index.class
```

Normally you, will find the source and data files in `/home/cs-301/Labs/Lab03`

1. A refresher of some `Collection<E>` methods... Develop a main method where two `ArrayList` are created, containing `String` and `Integer` elements. In the same main, for each list,
   **a.** Add three elements
   **b.** Delete the element with index 0
   **c.** Display the list.

2. Create and compile the two classes:
   **a.** `class ThingsA<T> implements Collection<T>`
   **b.** `class ThingsB<T> extends AbstractCollection<T>`
   Be minimalist, writing stubs for methods. Can you tell in advance which of the two will cause you less headaches? That is, the one you should attempt if you need to get either compiling as soon as possible. Make sure you look at the API, for the interface and the class.

3. Consider the class `Index`, which consist of constructors and a pair of integers. Here is the application interface:

```
public int ix, iy      // public fields
public Index(Index x)  // copy constructor
public Index()         // no arg,  makes both integers 0
public Index(int ix ,int iy)  // initializes
```

   The class `DataStruct` has as an array of `char` an initial index. The exercise consist in providing two inner classes implementing `Iterator<Index>`.. In addition, you may need create methods to run (testing the iterators).

**a.** . The first iterator should be called `Horizontal`. When `next()` is called the current index is retrieved, (provided it is good) and it is properly incremented. You may want to provide a constructor for it with the dimension of the array (in this particular case 4). So the sequences of indice by iterator loop is
$00, 01, 02, 03, 10, 11, 12, 13, 20 \ldots$

**b.** . The second iterator `Diagonal`, is very similar but the indices retrieved are
$00, 10, 01, 20, 11, 02, 30 \ldots$ Test your iterators by printing the char in a single line.

4. The inner class `Forward` implemented a forward traversing of the data structured defined by the weekly program `TriangleDS.java`. **a.** Implement a `Backward` iterator that traverses the structure backwards. **b.** Implement another iterator that traverses in some unconventional way, like down-and-left.