COMP 196ABL – Project #11 – LAST PROJECT
8 points with option of 4 extra credit points; due date – 8 am on Monday, May 6, 2019
Email your Java source files to: sgs@csun.edu

Reversing Link Order in a Linked List

Taking the base code available in the week14 folder on box.com for the classes FibList3, Int and Link, write a method called reverseList() that would appear in the file/class FibList3, parallel to the "main" method. This new method takes as its single parameter a reference to a linked list based on the class Link and it returns the same Link objects relinked in reverse order.

The output from the modified FibList3 program will be:

```
377 233 144 89 55 34 21 13 8 5 3 2 1 1 0
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

The second line resulting from the objects listed on the first line having been relinked in reverse order.

The code to implement reverseList() may reference only the class Link and not Int. This is to make the code portable which is the reason for have the two classes Int and Link rather than have these data structured merged as one class. By restricting reverseList() to just the class Link, we could copy the code for reverseList() to another program, such as the linked list of employees, without needing to change any of the code.

Remember that the objects are not to be recreated; the existing objects are to be relinked.

A non-recursive, loop-based solution is fine for the base 8 points and it is recommended you initially develop such a solution to gain an understanding of the linking structure. To receive the 4 extra credit points, design a solution that uses recursion rather than conventional looping statements. A recursive solution may consist of more than one method.