**CS 205**

**Lab 3 – More on Linked Lists**

**Objectives:**

Learn How to Implement Iterable and Iterator interfaces for Doubly Linked List

Learn How to Traverse Doubly Linked List outside its class (using the Iterator)

**Examples:**

1. Class **DLL** has been updated to implement the Iterable interface. Study it carefully to understand the implementation.
2. Class **TestIntegerDLL** example has been updated to include methods that returns the ***sum*** of the element in the list using the iterator directly, sum1(), and using the for-each loop, sum2().
3. Class **TestItemDLL** has been updated to print all items whose price is greater than a specified amount.

**Tasks:**

1. Update **TestIntegerDLL** to include a method that uses Iterator (or for-each loop) to print all the odd numbers in the list. Test your method by adding an option in the main method.
2. Update the **TestItemDLL** to include a method:

public static double totalCost(String itemName, int quantity, DLL<Item> list)

That uses Iterator (or foreach loop) to locate a given item, then computes and returns the total cost, given the quantity purchased. Test your method by adding an option in the main method.

1. Update your TestStudentDLL of Lab#2 to include a method:

public static void printGoodStudents(DLL<Student> list)

that uses Iterator (or foreach loop) to print all students in the list whose GPA is greater than 2.0. Test your method by adding an option in the main method.

1. Write an inner class, **DLLReverseIterator**<T>, inside the **DLL** class, similar to **DLLIterator**<T> but which iterates the elements of the list in reverse order. Update the ***iterator()*** method to return an object of this class instead. Test your new Iterator class using any of the Test classes. Example, test your task 1 above with the new iterator.