Exercise 16 (Part A) – The Big Three

Due Date: Today at the end of class

How to turn in your work:

1) Write the **name header** in the **DLL_ChildClass.cpp**
2) Print the **DLL_ChildClass.cpp** file ➔**FIT THE CODE IN ONE PAGE**
3) Turn in:
    a. Printed copy
    b. Project named **A250_E16_A_YourLastName_YourFirstName**

Project **ex_16a_the_big_three** contains a **Main.cpp** file with testing cases, the class **Node** already completed, and the class **DoublyList** also completed. The class **DoublyList** creates doubly-linked lists by inserting nodes to the back of the list. All functions in this class are already implemented, and the **member variables** are declared as **protected**.

You will be working in the **DLL_ChildClass** that **inherits** from the **DoublyList** class. Since the **DoublyList member variables** are **protected**, instead of being *private*, they can be directly accessed. A **constructor** and a **destructor** are already provided and need to be left empty (since there are **no** member variables for this class).

Your job is to implement the following functions in the class **DLL_ChildClass**:

- **Copy constructor**
    - The **copy constructor** is an **overloaded constructor** that differs from the default constructor by having as a **parameter** another object of the same class. Since it is a *constructor*, you will need to **initialize the calling object** just as you would in the default constructor.
    - Traverse the list passed by the parameter and copy each item of that list into the calling object.

- **Overloaded assignment operator**
    - Implement an **IF/ELSE** statement that checks whether the calling object and the parameter are the same list. If they are, output the error message, "Attempted assignment to itself." If the two objects are different, call the function **destroyList** to empty the calling object and then traverse the parameter list to copy each item into the calling object.

IMPORTANT: The purpose of this assignment is for you to **really** understand the **difference** between a **copy constructor** and an **assignment operator**. Make sure this is clear, because you will need to answer questions on the final exam about the **big three**.

(See expected output on next page.)

The **output** should be similar to the one below. Obviously, the addresses will differ; therefore, you will need to check that the lists are different.