Muzaffar Sharapov

Part C

I mostly replaced shared ptr in classes Node.cpp(changed from raw node to shared_ptr in lines: 27,48,65), Node.h(changed from raw node to shared_ptr in lines: 23,30,35,41), LinkedBag340(changed from raw node to shared_ptr in lines: 13, 14,18,24,41,62,128,145,159,167,181, changed from raw node to unique_ptr in lines: 38),LinkedBag.cpp(changed from raw node to shared_ptr in lines: 23,34,41,71,81,96,102,135,155,157,), LinkedBag.h(changed from raw node to shared_ptr in lines: 35,36,54,60). It's better to use smart pointers as smart pointers
In class Linked340.cpp I mostly used shared_ptr instead of raw node. But at the end I realized I could have used more unique_ptr(unique_ptr are smaller, and faster than shared_ptr.
) and use std:: move to convert unique_ptr to shared_ptr without taking over ownership.
I really wanted to use auto for weak_ptr in methods that use iteration but as I remember from the lecture we should not use auto weak_ptr and especially deprecated auto_ptr. I tried to use weak_ptr and shared_ptr for methods that use iteration; however I could not solve it without using auto. It shows error even though I followed the same format from the package, mickey and minnie exemple.

1. **In addition, please update and add destructor(s) so that the program displays more information *(in addition to the output required and described above)* when object(s) get destroyed.**

   ~ is a destructor.

   In Node.cpp class i added

```
Node<ItemType>::~Node() {

    cout << " 'Destructor is called' ";

}
```