**Code Reflection**

Writing the code for this exercise was straightforward. There were many hints in the comments that guided us through the coding process. The two issues that I ran into were in using the IDE (Visual Studio.) The provided zipped directory included two files named “LinkedList.cpp” which understandably kept giving errors. To resolve this, I simply renamed the extraneous file. The other issue was when writing the PrintList method. For some reason, the debugger was giving an error that “cout” was ambiguous. To resolve this, I explicitly declared it using “std::cout.”

**Pseudocode**

The pseudocode was essentially provided in the starter code. I will reinterpret it here.

LinkedList::LinkedList()  
Set head to null  
Set tail to null  
  
LinkedList::Append(Bid bid)  
Create new node *newNode*  
Set *bid* of *newNode* to the passed argument  
Set *next* of *newNode* to null  
If head is null, then set head and tail to *newNode*Else, set *next* of tail to *newNode*, set tail to *newNode*  
Increase size count  
  
LinkedList::Prepend(Bid bid)  
Create new node *newNode*  
Set *bid* of *newNode* to the passed argument  
Set *next* of *newNode* to null  
If head is not null, then set *next* of *newNode* to head  
Set head to *newNode*  
Increase size count  
  
LinkedList::PrintList()  
Create new node *current*  
Set *current* to head  
While *current* is not null:  
 print bidId, title, amount, and fund  
 set *current* to next

LinkedList::Remove(string bidId)  
If bidId in head is equal to bidId argument, then  
 set head to next  
 decrease size count  
 return  
  
Create node *current* and set to head  
Create node *temp*While current is not null:  
 If the next bidId is equal to argument, then  
 set *temp* to next  
 set next in *current* to next in *temp* delete temp  
 decrease size count  
 return  
 set current to next  
  
LinkedList::Search(string bidId)  
Create empty Bid *bid*  
If bidId in head is equal to argument, then return bid  
Create node *current* and set to head  
While current is not null  
 If bidId is equal to argument, then return bid  
 Else, then set current to next  
  
return (empty) bid (if search turns up blank)