

Assignment 1

Lab Explanation

The requirements for this assignment were to create a program that would return the count of the number of items in a linked list – excluding the head.

Code

```
C++ Kieran_Llarena_ListLength.cpp Lab8 2 X
C++ Kieran_Llarena_ListLength.cpp > main()
1  #include <iostream>
2  using std::cout, std::cin;
3
4  class IntNode {
5  public:
6      IntNode(int dataInit);
7      int GetNodeData();
8      IntNode* GetNext();
9      void InsertAfter(IntNode* newNode);
10
11 private:
12     int dataVal;
13     IntNode* nextNodePtr;
14 };
15
16 IntNode::IntNode(int dataInit) {
17     this->dataVal = dataInit;
18     nextNodePtr = nullptr;
19 }
20
21 int IntNode::GetNodeData() {
22     return this->dataVal;
23 }
24
25 IntNode* IntNode::GetNext() {
26     return this->nextNodePtr;
27 }
28
29 void IntNode::InsertAfter(IntNode* newNode) {
30     IntNode* tempNext = this->nextNodePtr;
31     this->nextNodePtr = newNode;
32     newNode->nextNodePtr = tempNext;
33 }
34
35 int GetCount(IntNode* headNode) {
36     int count = 0;
37
38     IntNode* currNode = headNode->GetNext();
39     while(currNode != nullptr) {
40         ++count;
41         currNode = currNode->GetNext();
42     }
43
44     return count;
45 }
46
47 int main() {
48     IntNode* headNode = new IntNode(-1);
49     IntNode* currNode;
50     IntNode* lastNode;
51
52     lastNode = headNode;
53
54     int maxCount = 0;
55     cout << "Enter the number of items to add:" << '\n';
56     cin >> maxCount;
57
58     for(unsigned int i = 0; i < maxCount; ++i) {
59         currNode = new IntNode(i);
60         lastNode->InsertAfter(currNode);
61         lastNode = currNode;
62     }
63
64     // this is called a linked list
65     cout << "There are " << GetCount(headNode) << " item(s) in this linked list" << '\n';
66
67     return 0;
68 }
```

The way that my code works is that it iterates through the linked list one by one using a while loop. This while loop will increment a counter variable that is declared at the beginning of the method. This loop will terminate once it detects that the next item of the linked list does not exist and then will return the count variable.

Test 1

```
./"Kieran_Llarena_ListLength"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_ListLength"  
Enter the number of items to add:  
3  
There are 3 item(s) in this linked list  
○ MacBook-Air-5:output kllarena$
```

Test 2

```
./"Kieran_Llarena_ListLength"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_ListLength"  
Enter the number of items to add:  
0  
There are 0 item(s) in this linked list  
○ MacBook-Air-5:output kllarena$
```

Test 3

```
./"Kieran_Llarena_ListLength"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_ListLength"  
Enter the number of items to add:  
15  
There are 15 item(s) in this linked list  
○ MacBook-Air-5:output kllarena$
```

Test 4

```
./"Kieran_Llarena_ListLength"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_ListLength"  
Enter the number of items to add:  
1000  
There are 1000 item(s) in this linked list  
○ MacBook-Air-5:output kllarena$
```

Assignment 2

Lab Explanation

The requirements for this assignment were to create an app that would locate the index of a value in a linked list.

Code

```
C++ Kieran_Llarena_ListLength.cpp Lab 8 C++ Kieran_Llarena_FindIndex.cpp Lab 1 X
C++ Kieran_Llarena_FindIndex.cpp > @ GetNodeData()
1 #include <iostream>
2 using std::cout, std::cin;
3
4 class IntNode {
5 public:
6     IntNode(int dataInit);
7     int GetNodeData();
8     IntNode* GetNext();
9     void InsertAfter(IntNode* newNode);
10
11 private:
12     int dataVal;
13     IntNode* nextNodePtr;
14 };
15
16 IntNode::IntNode(int dataInit) {
17     this->dataVal = dataInit;
18     nextNodePtr = nullptr;
19 }
20
21 int IntNode::GetNodeData() {
22     return this->dataVal;
23 }
24
25 IntNode* IntNode::GetNext() {
26     return this->nextNodePtr;
27 }
28
29 void IntNode::InsertAfter(IntNode* newNode) {
30     IntNode* tempNext = this->nextNodePtr;
31     this->nextNodePtr = newNode;
32     newNode->nextNodePtr = tempNext;
33 }
34
35 int IndexOf(IntNode* headNode, int targetValue) {
36     int size = 0;
37
38     IntNode* currNode = headNode->GetNext();
39     while(currNode != nullptr) {
40         ++size;
41         currNode = currNode->GetNext();
42     }
43
44     if(targetValue > size) {
45         return -1;
46     }
47
48     int index = 0;
49     if(size > 0) {
50         IntNode* currNode = headNode->GetNext();
51         while(currNode->GetNodeData() != targetValue) {
52             ++index;
53             currNode = currNode->GetNext();
54         }
55     } else {
56         return -1;
57     }
58
59     return (index - 1);
60 }
61
62
63 int main() {
64     IntNode* headNode = new IntNode(-1);
65     IntNode* currNode;
66     IntNode* lastNode;
67
68     lastNode = headNode;
69
70     unsigned int maxCount = 0;
71     cout << "Enter the number of items to add:" << '\n';
72     cin >> maxCount;
73
74     for(unsigned int i = 0; i < maxCount; ++i) {
75         currNode = new IntNode(i);
76         lastNode->InsertAfter(currNode);
77         lastNode = currNode;
78     }
79
80     int targetValue;
81     cout << "Enter the number that you want to target:" << '\n';
82     cin >> targetValue;
83
84     cout << "The index of the value " << targetValue << " is " << IndexOf(headNode, targetValue) << '\n';
85
86     return 0;
87 }
```

My code does the same thing as my code in the previous assignment. However, this time it returns the value of the index that the target value is located in. There are additional safety features to check if the target value is out of range or if the linked list size is 0.

Test 1

```
./"Kieran_Llarena_FindIndex"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_FindIndex"  
Enter the number of items to add:  
5  
Enter the number that you want to target:  
3  
The index of the value 3 is 2
```

Test 2

```
./"Kieran_Llarena_FindIndex"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_FindIndex"  
Enter the number of items to add:  
0  
Enter the number that you want to target:  
10  
The index of the value 10 is -1  
● MacBook-Air-5:output kllarena$
```

Test 3

```
./"Kieran_Llarena_FindIndex"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_FindIndex"  
Enter the number of items to add:  
10  
Enter the number that you want to target:  
1  
The index of the value 1 is 0  
● MacBook-Air-5:output kllarena$
```

Test 4

```
./"Kieran_Llarena_FindIndex"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_FindIndex"  
Enter the number of items to add:  
10  
Enter the number that you want to target:  
100  
The index of the value 100 is -1  
● MacBook-Air-5:output kllarena$
```

Test 5

```
./"Kieran_Llarena_FindIndex"  
● MacBook-Air-5:output kllarena$ ./"Kieran_Llarena_FindIndex"  
Enter the number of items to add:  
100  
Enter the number that you want to target:  
77  
The index of the value 77 is 76  
● MacBook-Air-5:output kllarena$
```

