Objects and Classes

Assignment 1: List Count

Given the IntNode class, define the GetCount() function that returns the number of items in the list not including the head node.

// Constructor

Example 1: If the list contains: head -> 14 -> 19 -> 4

GetCount(HeadNode) returns 3

Example 2: If the list contains: head ->

GetCount(HeadNode) returns 0

```
IntNode::IntNode(int dataInit) {
                                                         this->dataVal = dataInit;
                                                         nextNodePtr = nullptr;
                                                                                                            // Return number of nodes in a list
€ main.cpp 1 ×
                                                 30
                                                                                                           int GetCount(IntNode* headNode) {
                                                 31
54
                                                      // Get node value
     #include <iostream>
                                                                                                      55
                                                                                                              /* Type your code here. */
                                                      int IntNode::GetNodeData() {
      using namespace std;
                                                                                                      56
                                                         return this->dataVal;
                                                                                                      57
      class IntNode {
                                                                                                      58
                                                                                                            int main() {
      public:
                                                                                                      59
                                                                                                               IntNode* headNode = new IntNode(-1);
                                                      // Get pointer to next node
         // Constructor
                                                      IntNode* IntNode::GetNext() 
                                                                                                      60
                                                                                                               IntNode* currNode;
         IntNode(int dataInit);
                                                         return this->nextNodePtr;
                                                                                                      61
                                                                                                               IntNode* lastNode;
                                                 40
                                                                                                      62
         // Get node value
                                                                                                      63
                                                                                                               // Initiaize head node
 10
         int GetNodeData();
                                                      /* Insert node after this node.
                                                                                                      64
                                                                                                               lastNode = headNode;
 11
                                                         Before: this -- next
                                                                                                      65
 12
         // Get pointer to next node
                                                 44
                                                         After: this -- node -- next
 13
         IntNode* GetNext();
                                                                                                      66
                                                                                                               // Add nodes to the list
 14
                                                                                                      67
                                                                                                              for (int i = 0; i < 20; ++i) {
                                                      void IntNode::InsertAfter(IntNode* newNode) {
 15
         /* Insert node after this node.
                                                                                                      68
                                                                                                                  currNode = new IntNode(i);
                                                         IntNode* tempNext = this->nextNodePtr;
 16
            Before: this -- next
                                                                                                      69
                                                                                                                  lastNode->InsertAfter(currNode);
                                                         this->nextNodePtr = newNode;
 17
            After: this -- node -- next
                                                         newNode->nextNodePtr = tempNext;
                                                                                                      70
                                                                                                                  lastNode = currNode;
 18
                                                 50
                                                                                                      71
 19
         void InsertAfter(IntNode* newNode);
                                                                                                      72
 20
                                                      // Return number of nodes in a list
                                                                                                      73
                                                                                                               cout << GetCount(headNode) << endl;</pre>
 21
      private:
                                                      int GetCount(IntNode* headNode) {
                                                                                                      74
 22
         int dataVal;
                                                                                                      75
                                                                                                               return 0;
         IntNode* nextNodePtr;
                                                         /* Type your code here. */
                                                                                                      76
                                                                                                      77
```

Assignment 1 Tests:

Apply the following 4 tests.

```
1. Unit test (3 points)

Test GetCount() returns 3 for list with three items

Show details 

2. Unit test (3 points)

Test GetCount() returns 0 for an empty list

Show details 

3. Unit test (2 points)

Test GetCount() returns 15 for list with 15 items

Show details 

4. Unit test (2 points)

Test GetCount() returns 1000 for list with 1000 items

Show details 

Show details 

Y
```

Assignment 2: Index of list item

Given the IntNode class, define the IndexOf() function to return the index of parameter target or -1 if not found.

Note: The first index after the head node is 0.

Example 1: If the list contains: head -> 14 -> 191 -> 22 -> 99

IndexOf(headNode, 22) returns 2.

Example 2: If the list contains: head ->

IndexOf(headNode, 22) returns -1.

```
// Constructor
                                                                                                          // Return index of target item
     #include <iostream>
                                                                                                          int IndexOf(IntNode* headNode, int target) {
                                                       IntNode::IntNode(int dataInit) {
     using namespace std;
                                                                                                             /* Type your code here. */
                                                          this->dataVal = dataInit;
                                                          nextNodePtr = nullptr;
     class IntNode {
                                                  30
                                                                                                      56
     public:
                                                  31
                                                                                                     57
         // Constructor
                                                                                                          int main() {
                                                       // Get node value
         IntNode(int dataInit);
                                                       int IntNode::GetNodeData() {
                                                                                                             IntNode* headNode = new IntNode(-1);
                                                                                                             IntNode* currNode;
                                                          return this->dataVal;
        // Get node value
                                                  35
                                                                                                             IntNode* lastNode;
         int GetNodeData();
10
                                                  36
                                                                                                     62
11
                                                       // Get pointer to next node
                                                                                                             // Initiaize head node
12
         // Get pointer to next node
                                                                                                             lastNode = headNode;
                                                       IntNode* IntNode::GetNext() {
         IntNode* GetNext();
13
                                                          return this->nextNodePtr;
14
                                                  40
                                                                                                             // Add nodes to the list
15
        /* Insert node after this node.
                                                  41
                                                                                                             for (int i = 0; i < 20; ++i) {
16
            Before: this -- next
                                                                                                                currNode = new IntNode(i);
                                                       /* Insert node after this node.
17
            After: this -- node -- next
                                                          Before: this -- next
                                                                                                     69
                                                                                                                lastNode->InsertAfter(currNode);
18
                                                                                                                lastNode = currNode;
                                                          After: this -- node -- next
19
         void InsertAfter(IntNode* newNode);
20
                                                                                                     72
                                                       void IntNode::InsertAfter(IntNode* newNode)
21
                                                  47
      private:
                                                                                                             cout << IndexOf(headNode, 15) << endl;</pre>
                                                          IntNode* tempNext = this->nextNodePtr;
         int dataVal;
22
                                                          this->nextNodePtr = newNode;
                                                                                                     75
23
         IntNode* nextNodePtr;
                                                          newNode->nextNodePtr = tempNext;
                                                                                                             return 0;
                                                  50
                                                                                                      76
```

Assignment 2 Tests

Apply the following 5 tests.

```
1. Unit test (2 points)

Test IndexOf() returns 2 for target in 3rd position

Show details 

2. Unit test (2 points)

Test IndexOf() returns -1 for an empty list

Show details 

3. Unit test (2 points)

Test IndexOf() returns 0 for target in 1st position of list

Show details 

4. Unit test (2 points)

Test IndexOf() returns -1 if target not found

Show details 

5. Unit test (2 points)

Test IndexOf() returns 76 for 77th time in list

Show details 

5. Unit test (2 points)

Test IndexOf() returns 76 for 77th time in list
```

Submissions

Note: Do not forget to submit all two assignments and corresponding test outputs to receive full credit.

- 1 Name your C++ files FirstName_Lastname_ListLength.cpp, FirstName_Lastname_FindIndex.cpp.
- 2 Prepare your report in docx or pdf format and name it Firstname_Lastname.docx or Firstname_Lastname.pdf. Put both your assignments and corresponding tests in ONE report file.
- 3 Add the screenshot of your code to the report. All tests should be performed and the result screenshot be included in the report.

Note: Make sure to have your report containing both explanatnations and screenshots.