Practice Exercise

Linked List Code

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Instructions

Assume you are implementing the **function** createList as a **member** of the class **AnyList**.

Write the statement(s) needed as instructed in the questions.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 1

Declare a pointer **pNode** that points to objects of class **Node**.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 1

Declare a pointer **pNode** that points to objects of class **Node**.

```
Node *pNode;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 2

Use the **default constructor** to create a new node and make **pNode** point to it.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 2

Use the **default constructor** to create a new node and make **pNode** point to it.

```
pNode = new Node;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 3

What is the value of the member variable **data** stored in the node pointed by **pNode**?

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 3

What is the value of the member variable **data** stored in the node pointed by **pNode**?

The value is 0, because the default constructor sets the value of the variable **data** to 0.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 4

Store integer 2 in the node you just created.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 4

Store integer 2 in the node you just created.

```
pNode->setData(2);
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 5

Make the node you just created the first node in the list.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 5

Make the node you just created the first node in the list.

```
first = pNode;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 5

Make the node you just created the first node in the list.

Any member variables you need to update?

```
first = pNode;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 5

Make the node you just created the first node in the list.

Any member variables you need to update?

```
first = pNode;
++count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 6

Using pointer **pNode**, create another **Node**, but this time use the **overloaded constructor** and assign to the member variable **data** the value 4 and to the pointer **next** the value NULL.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 6

Using pointer **pNode**, create another **Node**, but this time use the **overloaded constructor** and assign to the member variable **data** the value 4 and to the pointer **next** the value NULL

```
pNode = new Node(4, NULL);
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 7

Make the first node of your list point to the newly created node, and then make **pNode** point to the first node.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 7

Make the first node of your list point to the newly created node, and then make **pNode** point to the first node.

```
first->setNext(pNode);
pNode = first;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 7

Make the first node of your list point to the newly created node, and then make **pNode** point to the first node.

Aren't you forgetting something?

```
first->setNext(pNode);
pNode = first;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 7

Make the first node of your list point to the newly created node, and then make **pNode** point to the first node.

Aren't you forgetting something?

```
first->setNext(pNode);
pNode = first;
++count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

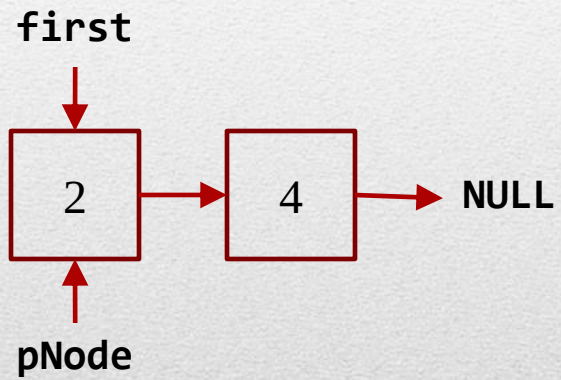
Question 8

Draw the list you have created so far.

Question 8

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Draw the list you have created so far.



```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 9

Move pointer **pNode** forward to the next node.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 9

Move pointer **pNode** forward to the next node.

```
pNode = pNode->getNext();
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 10

Using only pointer **pNode** add a new node to the back of the list, storing the value 6, and increment the member variable member count.
Use only a **total of 2 statements**.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 10

Using only pointer **pNode** add a new node to the back of the list, storing the value 6, and increment the member variable member count.
Use only a **total of 2 statements**.

```
pNode->setNext(new Node(6, NULL));
++count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 11

Using only pointer **pNode** add three additional nodes to the list, so that the list becomes:

first➔ 2 ➔ 4 ➔ 6 ➔ 3 ➔ 8 ➔ 9

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 11

Using only pointer **pNode** add three additional nodes to the list, so that the list becomes:

first➔ 2 ➔ 4 ➔ 6 ➔ 3 ➔ 8 ➔ 9

```
pNode = pNode->getNext();
pNode->setNext(new Node(3, NULL));
++count;

pNode = pNode->getNext();
pNode->setNext(new Node(8, NULL));
++count;

pNode = pNode->getNext();
pNode->setNext(new Node(9, NULL));
++count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 12

Delete the node that stores 4 (second node), **without** creating additional pointers.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 12

Delete the node that stores 4 (second node), **without** creating additional pointers.

```
pNode = first->getNext();
first->setNext(pNode->getNext());
delete pNode;
pNode = NULL;
--count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 13

Now your list looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

Using a **while** loop, print all elements in the list, **except the last**, without creating any new pointers and **without** using the member variable count.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 13

Now your list looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

Using a **while** loop, print all elements in the list, **except the last**, without creating any new pointers and **without** using the member variable count.

```
pNode = first;
while (pNode->getNext() != NULL)
{
  cout << pNode->getData() << " ";
  pNode = pNode->getNext();
}
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 14

Your list still looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

To which node is pointer **pNode** pointing?

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 14

Your list still looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

To which node is pointer **pNode** pointing?

To the node that stores 9.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 15

Your list still looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

Delete the first node.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 15

Your list still looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

Delete the first node.

```
pNode = first;
first = first->getNext();
delete pNode;
pNode = NULL;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 15

Your list still looks like this:

first➔ 2 ➔ 6 ➔ 3 ➔ 8 ➔ 9

Forgot the count again?

```
pNode = first;
first = first->getNext();
delete pNode;
pNode = NULL;
--count;
```

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 16

After deleting the first node, your list looks like this:

first➔ 6 ➔ 3 ➔ 8 ➔ 9

In a **single statement**, add a new node in front of the list, making it the **first node** of the list. Your node should store the value 1. Use a **second statement** to update the member variable **count**.

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Question 16

After deleting the first node, your list still looks like this:

`first➔ 6 ➔ 3 ➔ 8 ➔ 9`

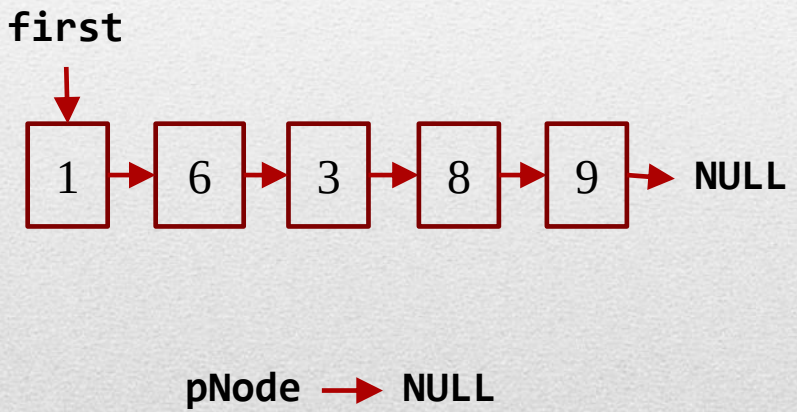
In a **single statement**, add a new node in front of the list, making it the **first node** of the list. Your node should store the value 1. Use a **second statement** to update the member variable **count**.

```
first = new Node(1, first);
++count;
```

Final List

```
class Node
{
public:
     Node() : data(0), next(NULL) {}
     Node(int theData, Node *newNext)
          : data(theData), next(newNext){}
     Node* getNext() const { return next; }
     int getData( ) const { return data; }
     void setData(int theData) { data = theData; }
     void setNext(Node *newNext) { next = newNext; }
     ~Node(){}
private:
     int data;
     Node *next;
};
class AnyList
{
public:
     ...
     void createList();
private:
     Node *first;
     int count;
};
```

Now your list looks like this:



Practice Exercise – Singly List Code (END)