Given the class **Node** and (partial) class **AnyList** below, write the **definition** of the **member** function **insertAndPrint** of the class **AnyList** that inserts a node to the front of the calling object and then prints the list starting from the second node up to the node before last. Make sure you add appropriate error messages when necessary.

```
class Node
{
public:
       Node() : data(0), ptrToNext(nullptr) {}
       Node(int theData, Node *newPtrToNext)
             :data(theData), ptrToNext(newPtrToNext){}
       Node* getPtrToNext() const { return ptrToNext; }
       int getData( ) const { return data; }
       void setData(int theData) { data = theData; }
       void setPtrToNext(Node *newPtrToNext)
             { ptrToNext = newPtrToNext; }
       ~Node(){}
private:
    int data;
    Node *ptrToNext;
};
```

```
class AnyList
{
public:
       …
       void insertAndPrint();
private:
       Node *ptrToFirst;
       int count;
};
```

SOLUTION:

```
void AnyList::insertAndPrint(int newItem)
{
      ptrToFirst = new Node(newItem, ptrToFirst);

      ++count;

      if (count < 3)
      {
            cerr << "There are less than three nodes in the list.";
      }
      else
      {
            Node *current = ptrToFirst->getPtrToNext();
            while (current->getPtrToNext() != nullptr)
            {
                  cout << current->getData()  << " ";
                  current = current->getPtrToNext();
            }
      }
}
```