**Lab 19 Building a linked list**

Start this lab with the code listed below. The LinkedList class defines the rudiments of the code needed to build a linked list of Node objects. You will first complete the code for its addFirst method. This method is passed an object that is to be added to the beginning of the list. Write code that links the passed object to the list by completing the following tasks in order:  
  
1. Create a new Node object.  
2. Make the data variable in the new Node object reference the object that was passed to addFirst.  
3. Make the next variable in the new Node object reference the object that is currently referenced in variable first.  
4. Make variable first reference the new Node.  
  
Test your code by running the main method in the LinkedListRunner class below. Explain, step by step, why each of the above operations is necessary. Why are the string objects in the reverse order from the way they were added?

public class LinkedList  
{  
 private Node first;  
  
 public LinkedList() { first = null; }  
 public Object getFirst()  
 {  
 if (first == null) { throw new NoSuchElementException(); }  
 return first.data;  
 }  
  
 public void addFirst(Object element)  
 {  
 // put your code here  
 }  
  
 public String toString()  
 {  
 String temp = "";  
 Node current = first;  
 while (current != null)  
 {  
 temp = temp + current.data.toString() + '\n';  
 current = current.next;  
 }  
 return temp;  
 }  
 class Node  
 {  
 public Object data;  
 public Node next;  
 }  
}  
  
---------------------------------  
  
public class LinkedListRunner  
{  
 public static void main(String[] args)  
 {  
 LinkedList myList = new LinkedList();  
 myList.addFirst("aaa");  
 myList.addFirst("bbb");  
 myList.addFirst("ccc");  
 myList.addFirst("ddd");  
 System.out.println(myList);  
 }  
}