The following code is a C++ example of a very limited implementation of a Linked List. It is written as a class wrapper for a Node structure and the functions to create a list, add values, and show the contents. Also note the destructor will eliminate the list elements from memory.

```
1.  class List {
2.      struct Node {
3.          int data;
4.          Node * next;
5.      };
6.
7.      Node * head;
8.
9.  public:
10.     List() {
11.         head = NULL;
12.     }
13.
14.     ~List() { // delete the list
15.         while(head != NULL) {
16.             Node * n = head->next;
17.             delete head;
18.             head = n;
19.         }
20.     }
21.
22.     void add(int value) {
23.         Node * n = new Node;
24.         n->data = value;
25.         n->next = head;
26.         head = n;
27.     }
28.
29.     void show(){
30.     while(head !=NULL) {
31.         Node * n = head->next;
32.         cout<< head->data <<", ";
33.         head = n;
34.         }
35.      cout << endl;
36.     }
37.
38. };
39.
```

Illustration 1: Linked List Class

Illustration 2 is a main program which adds 7 values to the list and then shows the contents.

```
1.  int main()  {
2.  time_t t=time(NULL);
3.  List MyList;
4.
5.  MyList.add(5);
6.  MyList.add(12);
7.  MyList.add(24);
8.  MyList.add(32);
9.  MyList.add(47);
10. MyList.add(54);
11. MyList.add(65);
12.
13. MyList.show();
14. cout<<"Today is: "<< ctime(&t) << endl;
15. return 0;
16. }
```

Illustration 2: Step 1. Main function

Step 1. Put the two pieces of code together in a Code::Blocks C++ console application and run the program to print out the results. Screen capture the output and turn for Step 1. Make sure the Time and Date are in the screen capture so it can be seen when reviewed.

Note: #include <> statements for libraries will have to be added plus “using std::” declarations.
#include <iostream>
#include <ctime>

Question 1. Why does the data print out in the sequence that it does?

Now modify the code by using a template for the List class. See the following:

Step 2. Modify the List class in Illustration 1 to be a templated class. Use the 'template' keyword to allow the class to accept any type of data for the list. Then use the following code for the main function and print out results. Screen capture the output and turn in for Step 2.

```
1.  int main()
2.  {
3.  time_t t=time(NULL);
4.  // create first list
5.  List<int> MyList;
6.
7.  MyList.add(5);
8.  MyList.add(12);
9.  MyList.add(24);
10. MyList.add(32);
11. MyList.add(47);
12. MyList.add(54);
13. MyList.add(65);
14.
15. cout << "Here is my first list " << endl;
16.  MyList.show();
17.
18. // Create second list
19. List<string> MyStringList;
20.
21. MyStringList.add("First String");
22. MyStringList.add("Second String");
23. MyStringList.add("Third String");
24. MyStringList.add("Fourth String");
25. MyStringList.add("Fifth String");
26.
27. cout << "Here is my Second list " << endl;
28. MyStringList.show();
29. cout<<"Today is: "<< ctime(&t) << endl;
30.   return 0;
31. }
```

Illustration 3: Step 2. Main function

Reference: http://stackoverflow.com/questions/397895/how-could-i-create-a-list-in-c