Lab 13 Study Questions

1. What is the purpose of an overloaded bracket ([ ]) operator?

2. Given a linked list of integers called L1, write a segment of C++ code that uses an Iterator to walk down the list and print out the values contained in the list.

3. Given a linked list of integers called L1, write a segment of C++ code that uses an index and the bracket operator to walk down the list and print out the values contained in the list.

4. What are the benefits and drawbacks to using the bracket operator?

5. What conditions might result in a Segmentation Fault in the index operator?

6. What conditions might result in a segmentation fault in the Iterator increment (++) , decrement (--) and dereference (*) operators?

7. What is the purpose of a try-throw-catch block?

8. A try may have more that one catch block associated with it. How does the compiler determine which catch it should use?

9. Why is exception handing used in modern programming languages?

10. How does an "lvalue" differ from an "rvalue"?

11. Given the following prototypes for an indexing operator, explain the functional difference between them.

   \[\text{LT operator} [\text{index}] \text{ const; and} \]
   \[\text{LT } \& \text{ operator} [\text{index}] \text{ const;}

12. Given the following prototypes for an iterator dereferencing operator, explain the functional difference between them.

   \[\text{LT operator} *() \text{ const; and} \]
   \[\text{LT } \& \text{ operator} *() \text{ const;}

13. Given the doubly linked list L:

LList2 <string> L;

And the following memory map for L:

```
address : 0x7ffd18e173c0   first : 0x1a28ed0   last : 0x1a28e80
    size : 9   direction : FORWARD
```

```
address : 0x1a28c50   data : this   prev : 0x1a28cf0   next : 0x1a28ca0
address : 0x1a28ca0   data : is   prev : 0x1a28c50   next : 0x1a28d40
address : 0x1a28cf0   data : a   prev : 0x1a28c90   next : 0x1a28c50
address : 0x1a28d40   data : doubly   prev : 0x1a28ca0   next : 0x1a28de0
address : 0x1a28d90   data : linked   prev : 0x1a28e30   next : 0x1a28cf0
address : 0x1a28de0   data : list   prev : 0x1a28d40   next : 0x1a28e80
address : 0x1a28e30   data : containing   prev : 0x1a28ed0   next : 0x1a28d90
address : 0x1a28e80   data : nine   prev : 0x1a28de0   next : 0
address : 0x1a28ed0   data : strings!   prev : 0   next : 0x1a28e30
```

a. What output is produced by the following statement if the output operator puts a single space between items in the list:

```cpp
cout << L << endl;
```

b. What does the ‘0x’ in each of the addresses represent?

c. Why is the first address significantly different from the rest of the addresses?