EECE.3220: Data Structures

Spring 2017

Lecture 19: Key Questions
March 8, 2017

1. (Review) Describe the design of the linked list class given below. What functions might you want to add to this class?

```
class LList {
public:
   LList();                                  // Default constructor
   LList(LList &orig);                       // Copy constructor
   ~LList();                                 // Destructor
   LList & operator=(const LList &rhs);      // Assignment
   bool isEmpty();                           // True if list is empty
   void display(ostream &out);               // Print contents
   void insert(int v);                       // Add new value to list
   void remove(int v);                       // Remove node with v

private:
   class Node {
   public:
      int val;       // Value in each node
      Node *next;    // Pointer to next node
   };

   Node *first; // Pointer to first node
};
```

2. Write definitions for each of the following functions:

```
// Default constructor
LList::LList()
{


}

// Copy constructor
LList(LList &orig)
{









}

//Destructor
LList::~LList() {









}
```

```
// Assignment operator
LList & LList::operator=(const LList &rhs) {









}

// True if list is empty
bool LList::isEmpty() {



}

// Print contents of list
void LList::display(ostream &out) {









}
```

```
// Add new value to list—assume list is maintained in order
void LList::insert(int v) {









}

// Remove node containing v, if it's in list
void LList::remove(int v) {









}
```

3. Explain why some linked list implementations maintain a head node that is either empty or maintains some information about the rest of the list.

4. Explain the design and usefulness of a circular linked list.

5. Explain the design and usefulness of a doubly-linked list.