CSC 212 Tutorial Solution
Lists & Double Linked Lists

Problem 1

```
public T mostFrequentElement() {
    T mfe = null;
    int max = 0;
    Node<T> p = head;
    while (p != null) {
        Node<T> q = p;
        int count = 0;
        while (q != null) {
            if (q.data.equals(p.data))
                count++;
            q = q.next;
        }
        if(count > max) {
            max = count;
            mfe = p.data;
        }
        p = p.next;
    }
    return mfe;
}
```

Problem 2

```
public static<T> void circularLeftShift(List<T> list, int n) {
    for (int i = 0; i < n; i++) {
        list.findFirst();
        T elem = list.retrieve();
        list.remove();
        while (!list.last())
            list.findNext();
        list.insert(elem);
    }
}
```

Problem 3

```
public void removeBetween(T e1, T e2) {
    Node<T> p = head;
    while ((p != null) && (!p.data.equals(e1)))
        p = p.next;
    if (p == null)
        return;
    Node<T> q = p.next;
    while ((q != null) && (!q.data.equals(e2)))
        q = q.next;
    if (q == null)
        return;
    p.next = q;
    q.previous = p;
    current = head;
}
```

Problem 4

```
public static <T> void reverseCopy(DoubleLinkedList<T> l1,
   DoubleLinkedList<T> l2) {
    if (l1.empty())
        return;
    while (!l1.last())
        l1.findNext();
    while (!l1.first()) {
        l2.insert(l1.retrieve());
        l1.findPrevious();
    }
    l2.insert(l1.retrieve());
}
```