CSCI 104L Lecture 4: Encapsulation and Classes

We will group together all data and functions that interact with that data into a common element, or `class`. This idea is called **encapsulation**. Here is an example class signature for a linked list of integers:

```cpp
class IntLinkedList {
    public:
        IntLinkedList ( );
        IntLinkedList ( int n );
        ~IntLinkedList ( );
        void prepend ( int n );
        void remove ( int toRemove );
        void printList ( );
        void printReverse ( );
    private:
        void printReverseHelper ( Item *p );
        Item *head;
};
```

Here is how we will implement `printReverse()`.

```cpp
void IntLinkedList :: printReverse ( ) {
    if ( head != NULL) printReverseHelper ( head );
}
void IntLinkedList :: printReverseHelper ( Item *p ) {
    if ( p->next != NULL) printReverseHelper ( p->next );
    cout << p->value ;
}
```

Here is a possible usage of our IntLinkedList:

```cpp
int main ( ) {
    IntLinkedList *myList = new IntLinkedList ;
    myList->printList ( );
    delete myList;
    return 0;
}
```

The constructor is necessary so that this doesn’t cause very bad behavior:

```cpp
IntLinkedList :: IntLinkedList ( ) : head ( NULL ) {}
You can have multiple constructors. You could allow a user to start a Linked List with a single node with value n.

```cpp
IntLinkedList::IntLinkedList (int n) : head(NULL) {
    prepend(n);
}
```

Both constructors can be used, one constructor does not replace the other:

```cpp
Int LinkedList *p = new IntLinkedList (), *q = new IntLinkedList(3);
```

A destructor is necessary to prevent memory leaks:

```cpp
IntLinkedList::~IntLinkedList () {
    Item *p = head, *q;
    while (p != NULL) {
        q = p->next;
        delete p;
        p = q;
    }
}
```

The following problem can occur; what is this mistake called?

```cpp
class Item {
    public:
        void setValue(int value);
    ...
    private:
        int value;
    ...
};

void Item::setValue(int value) {
    value = value;
}
```

Here is how to fix it:

```cpp
void Item::setValue(int value) {
    this->value = value;
}
```