Announcements

CS2100 - End of C++ and Simple Linked Lists
What does this do?

\[ c = 0 \]

Use c for something?

My float vec C:

Another issue:
return this;

for (int i = 0; i < size; i++)
  a[i] = other[i];

a = new float[size];

delete [] other;
size = other.size;

if (this.size == other.size)
  MyFloatVec operator=(const MyFloatVec & other);

Solution: rewrite the "$\neq C$" operation $x = y$ as $x \neq y$. \"C\"
- deep copies
- memory leaks

Why?

3. Operator

Destructor

Copy Constructor

Receipt: Freekeeping Functions
enum: user defined types

enum: Rainbow 3 color, Sparkler

enum: Color 4 color, Blue, Green

if (sky == "Blue")
    Rainbow menu = Sparkler;

Color grass = Green;

Color sky = Blue;

count = "It's nice out today!"

not: f (sky == "Blue")
String MyString;
bool isRegele;
MealType mealType;
String name;
String password;

Regular Kosher;

Ex: enum MealType { No, Purr, Veg }

Useful for simple collections of objects.
C or J: Not included

pass. receive = \textit{Koster}

\[ 12 \text{ or } 34 \text{ or } 5 \]

\[ \text{pass = } 3 \text{ or } \text{Smith} \text{ or } \text{true} \]

\text{Passenger}

\text{We can then create instances of a}\n
\text{Using such}
More Complex

```
P = new Passenger();

P.name = "Barbara Wright";
P = new Pref (P);
```

*(p) is Frequent Flyer No = false
name = "Barbara Wright"
P = new Pref (P);
3

```c
if (a > 10) {
    return 5;
} else {
    return a;
}
```

\[ T \in \min \{ (9, 9), (16) \} \]

Example: Type coercion

We can use templates like variable types. Multiple classes, e.g., int and floats. If we want a function to work for

\[ < \text{Templates} \]
or password

min (x, y)

exit 2 = min (x, y)

int x = 53;

t = y (96);

Important: Will not work for any class with appropriate operator.
input or strings or lists. Code will make a list of these work in classes, also.

Templates in classes
Using a template.

MyList < String > list2.

MyList < int > list1.
A collection of nodes that follows a linear ordering.

**Singly Linked List**
Can expand easily
- more flexible: no max size.

So why?

- Similar to arrays as a building block.
- In a lot of our data structures

This linked structure will show up

- Over is it like Python lists

We'll write a class

Note: This is not trees

Why this structure?
null

insert
remove
test if empty

function (int, string)

Snode

ptr node data

struct Snode

data + pointers

What is a node + how do we use code it?

Implementation
3.

```java
class Node {
    Object obj;
    Node next;
}

class LinkedList {
    Node head;

    public void print() {
        Node current = head;
        while (current != null) {
            System.out.println(current.obj);
            current = current.next;
        }
    }
}
```

```java
class Node {
    Object obj;
    Node next;
}

class LinkedList {
    Node head;

    public void print() {
        Node current = head;
        while (current != null) {
            System.out.println(current.obj);
            current = current.next;
        }
    }
}
```

```java
class Node {
    Object obj;
    Node next;
}

class LinkedList {
    Node head;

    public void print() {
        Node current = head;
        while (current != null) {
            System.out.println(current.obj);
            current = current.next;
        }
    }
}
```
public:

Functions (listed in o.h.

- void removeFront();
- void addFront(const & o.h.
  - const & o.h.
  - const & o.h.
  - const & o.h.
  - const & o.h.

- bool empty();
- size_t size();
- SingletonList & front();
- SingletonList & rear();
Next:

let's code it.

will post on test the on

schedule page.