RULE OF THREE
LINKED LISTS CONTD

Problem Solving with Computers-II

C++

```
#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```

Read the syllabus. Know what's required. Know how to get help.

CLICKERS OUT – FREQUENCY AC

Destructor

The destructor is invoked when the object is removed
from memory

```
void foo(){
    IntList list1;
    IntList *p = new IntList;
}
```

How many times is the destructor invoked for the above
code?
A. Never
B. Once
C. Twice

Copy constructor

```
IntList list1;            //default constructor is invoked
//Copy constructor is invoked in all cases below:

IntList list2(list1);
IntList list3 = list1;
IntList *p = new IntList(list1);
```

- The copy constructor creates and initializes a new instance to be the copy of another instance of the class
- A class always has a default copy constructor which may be overloaded
- Why overload the copy constructor

Overloading the copy constructor

- Which of the following classes that you have implemented is a good candidate to overload the copy constructor

A. Statistician class from PA1

B. IntList (implemented in class)

Copy assignment

```
IntList list1, list2;  //default constructor is invoked
//Copy assignment is invoked in all cases below:


list1 = list2;
```

- The copy assignment should result in list1 having a copy of the data of list2
- A class always has a default copy assignment which may be overloaded
- Why overload the copy assignment

RULE OF THREE

If a class defines one (or more) of the following it should probably explicitly define all three:

1. Copy constructor
2. Copy assignment
3. Destructor

Value semantics: Copy assignment and copy constructor

Value semantics means treated objects as values and creating copies on

- Copy assignment
- Copy constructor

What is the output of this code?

Next time

- Run time analysis