RULE OF THREE
LINKED LISTS CONTD

Problem Solving with Computers-II

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

CLICKERS OUT – FREQUENCY AC

Memory Leaks

- Data created on the heap with **new** must be deleted using the keyword **delete**
- **Code has a memory leak if**
  - Data on the heap is never deleted or
  - Pointer to the data is lost

- Use valgrind to detect leaks

- Code that results in a leak

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

```
./valgrind –leak-check = full <name of executable>
```

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. De-constructor

1. What is the behavior of default copy-constructor, copy-assignment and deconstructor (taking linked lists as example)?
2. When and why do we need to overload these methods?
3. What is the desired behavior of the overloaded methods for linked-lists?

De-constructor: Default behavior

```
void foo(){
    IntList ll;
    ll.insert(100);
    ll.insert(50);
     ll.insert(75);

}
```

```
class IntList{
public:
      IntList(){head = tail = nullptr;}
      void insert(int value);
private:
      //Definition of struct Node
      //not shown here
      Node* head;
      Node* tail;
};
```

Does the above code result in a memory leak?

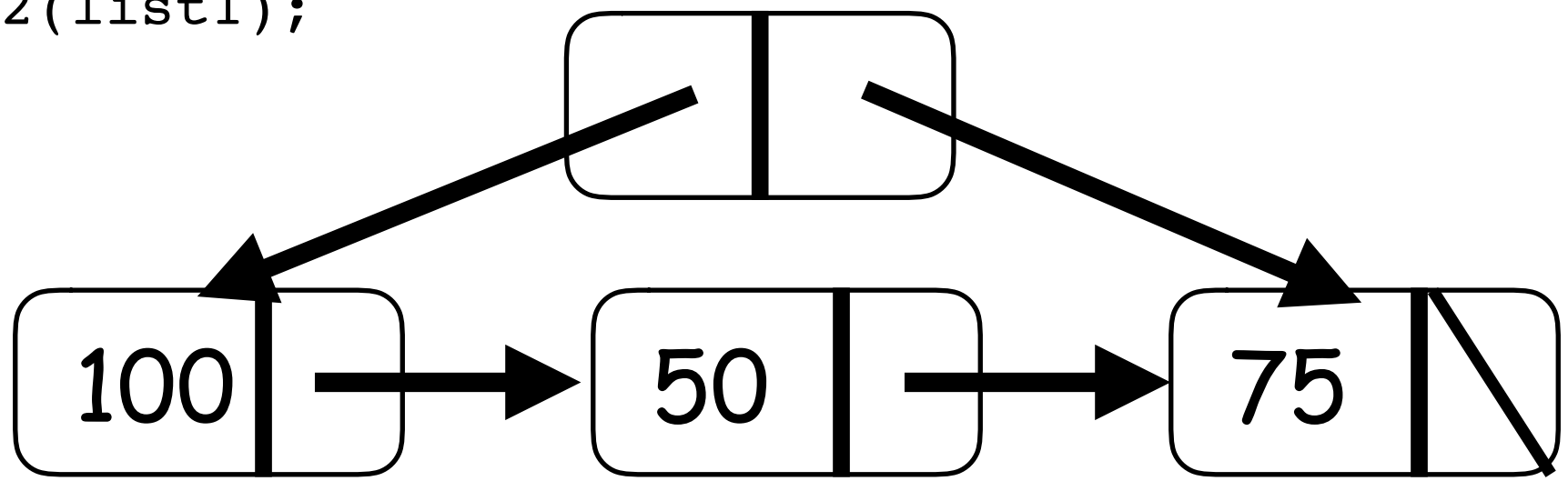
A. Yes
B. No

De-constructor: Default behavior

```
void foo(){
    IntList ll;
    ll.insert(100);
    ll.insert(50);
     ll.insert(75);
}
```

Copy constructor: Default behavior

```
void foo(Intlist list1){
    IntList list2(list1);

}
```



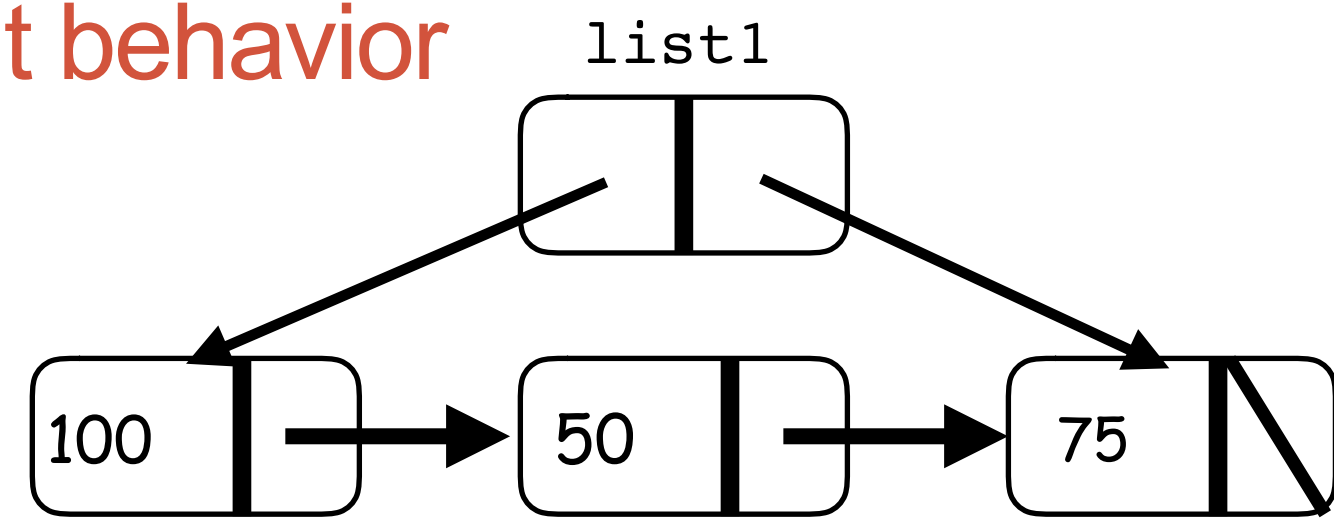
Copy assignment

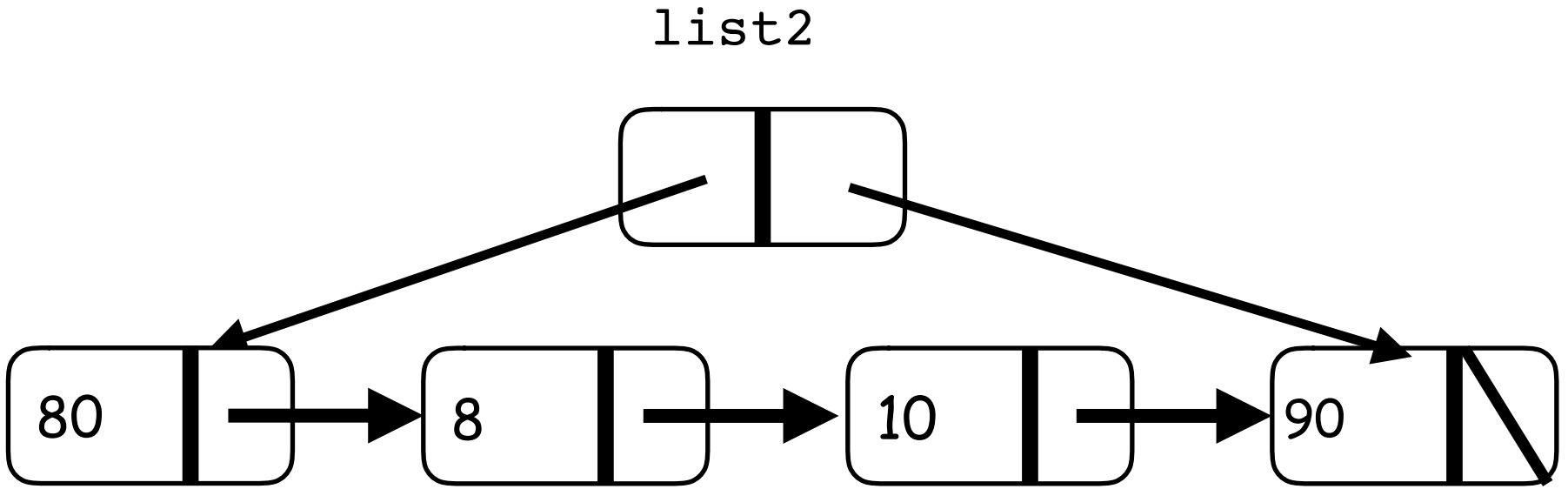
```
IntList list1, list2;  //default constructors called

list1 = list2; //Copy assignment is called
```

- 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?

Copy assignment: Default behavior

```
list2 = list1;
```





Value semantics: Copy assignment and copy constructor

Value semantics means passing objects to functions by value. The methods invoked are:

- Copy assignment
- Copy constructor

Next time

- Run time analysis