Wrapup: reference and value parameters.
Wrapup: Classes; pointers to them, the this pointer, member functions.

Character and C-string details and summary.

Doubly Linked List Programming, needed for Proj 2.

Studying/Dissection/Reverse Engineering.

Data structure editor Core object plus linked node class example:

CSI 310: Lecture 13
23.6; it was NOT 23.6.

The old value of HEAD was the ADDRESS of the node "housing"

```
(4) Copy the address of the node containing 14.7 into automatic variable HEAD. This operation OVERWRITES the previous value in HEAD, so it must be done AFTER (3).

(3) If TAIL==NULL, the original list was empty, so TAIL=HEAD gives TAIL its correct value!

old value before step 4

(2) Copy pointer value from HEAD into the TINK held of the new node. (This value is the address of the node containing 23.6.

(1) Run the dynamic memory allocator "new" of the C++ support library.

For both styles, the computer does the same things with the same results:

University at Albany Computer Science Dept.
meaning, plus practice solving problems!

understanding of variables, data and (C/C++) language

Have patience: Speed and elegance come from precise
these steps.

If it fails for boundary cases, try to fix it and repeat

(Empty: Empty: Empty: Empty: Empty)

Then, check it for any boundary case(s). (E. C: E: E: E:
and draft code if not.

Check that it works in the general case. Re-do design
idea in the general case.

Advice: Try to design and draft code for an algorithm
otherwise, it's good!

This last C++ statement works when the list was empty, and does nothing.
```c
{ } // edtorcore() : HEAD=TAIL=NULL; */ etc */
public:
node *TAIL;
node *HEAD;

private:
}
}
}
}
}
}
}
```
node *cutHEAD; // cutTAIL; // Other data members omitted

3. Cursor position is at the end of the buffer. // or last line. //

2. Cursor position is before the 2nd, 3rd, etc.,

(insert new lines at the beginning in all cases) //

1. Cursor position is the first buffer line

Your job: How (?) to distinguish

private:

Inserted at the beginning of the main list. //

post: the line referred to by pch has been

(2) pch holds the address of a dynamic allocate string.

pre: (1) this is properly constructed

void insertFront(char * pch) ;

private:
{ 
    return 
    { 
        TAIL = tp;
    }

    if (TAIL == NULL) 
    { 
        // If HEAD is a class data member
        tpp = tpp ? 
            tp->prevc = NULL; // NULL is a standard macro.
        // If HEAD is a class data member
        tp = new node;
        node *tp; // Or node *temp--node-pointer;
    }

    void editorcore::insertpron(char * pch) 
    { 
        #include <stdlib>
        // For NULL
        
        #include "editorcore.h"
        //editorcore.hxx
        // University at Albany Computer Science Dept.
structures and the working (faulty?) of the code on them.

(4) THINK/IMAGINE and ANALYZE by drawing BOUNDARY CASE data

(3) DRAW on the diagrams ONE-TO-ONE the effects of each code line in

(2) DIAGRAM one typical data structure for the argument to

(1) DIAGRAM one typical property constructed during code and its connected

Study/Dissecting/Reverse Engineering:
(i.e., executed return; or falls out.)

Automatically deallocated WHEN exitInterfaceOrientation

(i.e., called, invoked, etc.)

Automatically allocated WHEN exitInterfaceOrientation

is ACTIVATED

tp is a local storage extent variable

NOT the variable named tp

POINTS TO.
delete tp; deletes the OBJECT/STORAGE/VARIABLE the value of tp

Main and Savage warn "Don't code delete tp:"

University at Albany Computer Science Dept.