1 Students speak on coding style

'I understand the concept of proper indentation; are you really taking points off for that?'

'I didn't know you were grading for style; can I turn in another version with comments?'

2 How bad coding habits develop

- Enthusiasm for solving a problem overwhelms dim memories about lectures on coding style.
- Confidence in the understandability of their code makes documentation seem unimportant.
- Eagerness to get started sabotages intelligent planning of identifier names.
- Temporary changes which alter program structure become permanent—but indentation was never adjusted because the change was ‘temporary’.
- Plans to clean up later fall by the wayside as students move on to other tasks.
- A student’s natural desire to optimize leads them to do only what must be done: ‘Is this going to be on the test?’

3 Why this assignment helps

By forcing students to decipher a program written in poor style, the assignment makes them aware of good style’s importance.

4 The ugly code

```c
#include <stdio.h>

struct cf
{
    char ch;    /* ch is a character */
    int fr;     /* a number */
};

int process_data(char lc, struct cf cc[], int s);

void ss(struct cf cc[], int s);

main()
{
    int s = 0;
    struct cf cc[200];
    char lc;
    int i, l;

    /* Loop until end of file */
    while ((lc = getchar()) != EOF)
    {
        l = process_data(lc, cc, s);
        if (l > 0)
            cc[l].ch = lc;
        cc[l].fr++;
    }      /* Increment fr */
    else
    {
        cc[s].ch = lc;
        cc[s].fr = 1;
        s++;
    }

    /* Call ss function */
    ss(cc, s);

    /* Print answer */
    printf("Most frequent character: %c\n", cc[0].ch);
    return 0;
}

int process_data(char lc, struct cf cc[], int s)
{
    int i;
    for (i = 0; i < s; i++)
    {
        if (cc[i].ch == lc)
            return i;
    }
    return -1;
}

void ss(struct cf cc[], int s)
{
    int f, i, j;
    struct cf X;

    for (f = 0; f < s; f++)
    {
        /* Get next x */
        x = cc[f].fr;
        for (i = f + 1; i < s; i++)
        
            if (cc[i].fr > x)
            {
                y = i - 1;
                x = cc[i].fr;
            }
        if (x >= 1)
            X = cc[f];
        cc[f] = cc[y];
        cc[y] = X;
    }
}
```

5 What makes this program ugly?

1. Poor identifiers
   (a) Short, nondescriptive identifiers
   (b) Identifiers easily confused with numbers (e.g., lower-case l)
   (c) Identifiers easily confused with each other (e.g., x and X)
   (d) Vague function names

2. Poor comments
   (a) Lack of descriptive comments
   (b) Comments that state the obvious
   (c) Misleading comments

3. Poor layout
   (a) Misleading indentation
   (b) Poor placement of braces

6 The assignment

The students were divided into three groups. Each group was assigned one of these maintenance tasks:

**Group 1:** Modify the program so that it prints the most frequently occurring letter and the most frequently occurring digit. In both cases, print the character and the number of times it occurs in the input.

**Group 2:** Modify the program so that it prints the most and least frequently occurring characters in the input. In both cases, print the character and the number of times it occurs in the input. Do not count the characters that occur 0 times.

**Group 3:** Print every character that occurs 5 or more times in the input and the number of times the character occurs.