

C SC 397a, Spring 2010
Assignment 5
Due: Monday, March 1 at 22:00:00

Problem 1. (30 points) feature.txt

This problem was inspired by actual events at one of Tucson's largest software companies.

The program below, which can be found in `$FILES/a5/CharList.cc`, has a bug that's baffling the developer: The characters in the list print as expected in the debugging routine, but when the `at(int)` and `length()` methods are used in a loop, no output is produced.

If the developer had taken full advantage of a language feature already covered in our class, the "bug" would be quite apparent.

In a text file named `feature.txt`, answer these two questions: What's the language feature that the developer should have used? Exactly how would using that feature have made the bug apparent?

Here's the puzzling program:

```
#include <cstdio>
#include <cstring>
#include <cstdlib>

class CharList {
public:
    CharList(char* s) {
        itsChars = new char[strlen(s)+1];
        strcpy(itsChars, s);
    }

    //
    // Iterate over the characters, returning a NUL when done
    //
    char get() {
        if (*itsChars)
            return *itsChars++;
        else
            return 0;
    }
    //
    // Return the nth character
    //
    char at(int n) {
        if (n < length())
            return itsChars[n];
        else
            return 0;
    }
    int length() { return strlen(itsChars); }
private:
```

```

        char *itsChars;
    };

//
// debugging routine -- print the characters in the list
//
void print(const char *label, CharList& L)
{
    printf("%s", label);
    const char *fmt = "%c";
    while (char c = L.get()) {
        printf(fmt, c);
        fmt = ", %c";
    }
    puts("");
}

int main(int argc, char **argv)
{
    if (argc != 2) {
        printf(stderr, "Usage: %s STRING\n", argv[0]);
        exit(1);
    }

    CharList L(argv[1]);

    print("argv[1]: ", L);

    printf("Ready to enter loop...\n");
    for (int i = 0; i < L.length(); i++) {
        printf("in loop\n");
        printf("L[%d] = '%c'\n", i, L.at(i));
    }
    printf("Done with loop...\n");
}

```

Problem 2. (39 points) answers.txt

Create a text file named `answers.txt` with brief responses to the following questions/problems:

- (1) Is the `Counter` constructor shown at the top of slide 59 a default constructor? Justify your answer. (5 points)
- (2) A fellow student claims that there is a bug in `g++`, as follows: *"The definition `X **p = new X*[5];` causes five instances of `X` to be created, but the constructor of `X` is never called."* Explain what `new X*[5]` actually does. (5 points)

- (3) Here is a version of `Range::equal` that looks like it was written by a programmer who knows only a little C++:

```
int Range::equal(Range* a, Range* b)
{
    return a->span() == b->span();
}
```

Write a function named `equal_v2` that performs the same essential computation—comparing two `Ranges` for equal spans—but uses three useful C++ features that are not used above. (9 points)

- (4) Slide 75 poses this question: "What happens if the above definition of `getArea()` is placed in `Rectangle.cc` instead of `Rectangle.h`?" Answer that question. (5 points)
- (5) Can a static member function have the `const` qualifier introduced on slide 99? Defend your answer with a brief argument. (5 points)
- (6) When `$FILES/a5/copyctors.cc` is run, how many times is the copy constructor for `X` invoked? Defend your answer with a brief argument. Hint: If you spend more than five minutes on this problem you've found a hard way to do it! (5 points)
- (7) For five points, identify the most serious error in this class definition:

```
#include <cstring>
class X {
public:
    X(const char *name) { itsName = strdup(name); }
    ~X() { delete [ ] itsName; }
private:
    char *itsName;
};
```

If you're unfamiliar with `strdup(3)`, use `man strdup` to learn about it.

Problem 3. (Extra Credit) `extra.txt`

Submit a plain text file named `extra.txt` with the following.

- (a) (1 point extra credit) Estimate how long it took you to complete this assignment. Other comments about the assignment are welcome, too. I appreciate all feedback, favorable or not.
- (b) (1-3 points extra credit) Cite an interesting course-related observation (or observations) that you made while working on the assignment. The observation should have at least a little bit of depth.

Think of me saying "Good!" as one point, "Interesting!" as two points, and "Wow!" as three points. I'm looking for quality, not quantity.

Miscellaneous

Don't hesitate to ask me for hints and/or help if you have trouble with a problem.

Deliverables

Use `turnin` with the tag `397a_5` to submit your solutions for grading. The deliverables for this assignment are `feature.txt`, `answers.txt`, and if you choose to submit it, `extra.txt`.