

CS150 Assignment 6

Cheating Checker

Date assigned: Friday, November 13, 2009

Design Document Due: Monday, November 16, 2009, 8 pm (5 points)

Date due: Monday, November 23, 2009, 9:15am (45 points)

Total points: 50

Problem statement

This assignment involves writing a program that can aid in identifying students that have cheated on programming assignments by copying someone else's program and making only cosmetic changes to it. In order to perform this task, you will need to write a program that can analyze any C++ program. The analyzed program will be given an identification number to identify the program. This number will be constructed in such a manner that minor cosmetic changes to a program will result in a program that produces a number quite close to the number of the original program. Numbers from programs written by honest students should be quite different.

Use the following formula to produce the identification number:

$$\begin{aligned} \text{identification number} = & 5 * (\text{number of identifiers}) + \\ & 2 * (\text{number of integers}) + \\ & (\text{number of characters in all string literals}) + \\ & 4 * (\text{number of other symbols}) \end{aligned}$$

For this assignment, you will need to identify the number of identifiers, integers, characters in string literals, and other symbols where:

1. identifier - a letter followed by zero or more letters or digits (e.g. int, x1, main, cout)
2. integer - a digit followed by zero or more digits
3. literal - a character string enclosed in quote (") marks (e.g. "hi there")
4. other - any character other than those mentioned above, which are not:
 - (a) part of a comment or
 - (b) whitespace

Your program must output the number of identifiers, integers, total characters in all literals, comments, others, and the identification number.

You must use functions in this program. You must have at least five functions, one for each of the possibilities listed above (identifier, integer, literal, comment, and other). You will want to read the data file **one character at a time**. After identifying the character, call the appropriate function to find the end of the string of characters you are processing (i.e. the end of the identifier, the end of the integer, etc.).

Notes

1. Any symbol must begin and end on the same line.
2. Do not count the quote marks as characters in the literal string.
3. A comment is denoted by //. All comment characters are skipped over and not counted in calculating the identification number. The // comment is terminated by the end of line character.
4. Do not count whitespace as anything unless it is within quotes.
5. There is a C++ standard library called `<cctype>` that has useful character functions such as `isdigit`, `isalpha`, and `isspace`.
6. I will place a file called `06test.cpp` in the **CS150-02 Public** folder on Turing next week. I will be using this file as one of the test cases for your program.
7. Your program must open the file to be tested, find the correct number for each piece of data needed in the identification number, and then calculate the identification number. Your output is to look exactly like the following with correct values depending on the program tested.

```

*****
*                                     CHEATING CHECKER                                     *
*****

FILE TESTED: 06test.cpp

IDENTIFIERS  INTEGERS  CHARACTERS  OTHERS  IDENTIFICATION #
-----
              10          7          14          21          162

*****

```

This program is more difficult than previous assignments. See me early if you have questions or problems.

Also, when writing this program you should build and test as your code one piece at a time. For example, after you have written the function to find identifiers, test it on a sample file, then move on to the next function.

To complete this assignment you must

1. Create a new C++ project in Visual Studio. Name your project **06Checkerxxxxxxx**, where xxxxxxxx must be replaced by your PUNetID. As an example, my project would be called "06Checkingryandj". It is vital that you name your project correctly!

2. Type the solution (**fully documented/commented**) to the problem into your project.
3. Remember to enter in your name as the author of the program.
4. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly. Be aware that C++ is case-sensitive. Also, there must not be any warnings when compiling your program.
5. Once you are sure that the program works correctly, it is time to submit your program. You do this by logging on to Turing and placing your complete project folder in the **CS150-01 Drop** folder. Make sure that you copy your program folder and don't move the folder. If you move the folder, then you will not have your own copy!

Submit an electronic copy of your design document

Before you start you need to think about the data in your program and the calculations and loops you will need to perform. Answer the following questions in a **new** GoogleDoc (**CS150_06ProgramDesignPUNetID**) and share the document with the instructor (profchadd@gmail.com). Be sure to answer the questions in complete sentences where appropriate. This design document is due on **Monday at 8 pm**.

Design Questions:

1. List each function prototype for each function you are going to use in your program. After each function prototype, describe in English how the function is going to work.

As an example,

```
unsigned int sum (unsigned int low, unsigned int high);
```

This function will sum up all of the numbers between low and high inclusive using a for loop and return the sum.

This is a very simple example, but you get the idea. I should be able to take your English explanation and easily write the code.

2. Report on the value of each of the following for the program below:

Identifiers
Integers
Characters
Others
Identification Number

```

//*****
// Cheating Checker Test
//*****

#include <iostream>

using namespace std;

int main ()
{
    const double PI = 3.14159; // approximation for PI
    double radius;           // radius of a circle

    cout << "Enter a radius:";
    cin >> radius;

    cout << "Area is: " << PI * (radius * radius)
         << endl << endl;

    return 0;
}

```

Notes

1. You must follow the coding standards.
2. You must use constants when possible.
3. Your program will be graded on **efficiency and proper function use**.
4. You may only use the C++ programming concepts covered thus far in class. Do not use any more advanced concepts that we have not covered or any other programming concepts that you have had experience with.
5. Your output must look **exactly** like the sample given.
6. If this program sounds difficult, it's not that bad if you get an EARLY start. Make sure you understand all of the pieces before beginning to code your solution. Code your solution a piece (function) at a time not all at once. It makes for much smoother debugging.

Remember, this is an individual assignment. Refer to the syllabus for assignment policies