Homework 2

After unzipping the HW2.zip, you'll notice a subdirectory *userid*, and under that a directory *HW2Code* with some C++ code. This code will be used in steps A, B and C below. A Makefile is provided in the C directory where the Part C solution will go.

Part A.

Draw virtual function tables (VFTs) for Base and Derived, and arrows from the VFTs to the code executed. You can do this by hand if desired. These should be in a .pdf or JPEG file with the appropriate extension with the name of A.

Part B.

Examine the code in HW2Code. In the file HW2.cpp you will notice that each line has an Sx, where x is an integer, after it. You will need to put what is printed out by these lines for Part B.

You could compile and run the program but you won't learn anything, and C++ is incapable of learning, so there really isn't any value in that other than a few points and perhaps discovering on the test that you really don't know what is going on. Feel free to run it after putting what you think the output is into B.txt, to check and correct your results. If you go some wrong, figure out why and correct your results.

Fill in your answers in B.txt using an editor, by leaving a space after the ":" an putting your answer. Thus, in B.txt, for S0 you would have the line

S0: XYZ

If multiple lines are printed by a statement, answer like this (this isn't the correct answer, by the way):

19: Derived::bar
Base::f4

If a line gives a compile time error, answer "Err" (without the quote marks). For the rest of the program assume that the error line was not executed.

If a line prints nothing, and does not give an error, answer "Ok" (also without the quote marks).

Part C.

Delete the lines that cause an error in HW2.cpp in Part B. Then, by only changing the Derived class, make the call to f4() in S15 and S25 of main print "Derived::f4". Again, by only changing the Derived class, make the call in S19 and S29 print out

Derived::foo1
Derived::f1

Part D.

Write a program that does:

1. Defines a class Staticy that uses a static function and variable as described next.

2. Defines a static function makeStatic(int i) that calls the constructor for Staticy and returns a pointer to the newly created Staticy object.
3. Defines a private constructor that is called by makeStatic. The constructor takes a single integer argument.
4. Counts the number of times the private constructor is called with an even number as the argument.
5. Defines a static getter that returns the value of the count.
6. The main function (code is given in directory D) will call makeStatic and print out the value of the count.

How to turn in your homework:

Use the same directory structure as exists under the HW2 directory. That is, you should have a *userid* directory. Inside that directory there should be a file A.pdf (or A.jpg or A.jpeg) file with the answer to A and a file B.txt with the answer to B. Also in the *userid* directory there will be a *C* directory that contains the code for Part C and a D directory that contains the code for part D. You do not need to include a HW2Code directory. You should not have any executables in what you turn in. Zip up and turn in *userid.zip.*