

Name: _____

Problem	Points	Score
1	50	
2	50	
Total	100	

Notes:

- (1) Please see the instructions sent in email to the class for how to submit your work.
- (2) You are allowed to use all the web resources at your disposal except other human beings (and talking to someone via a chat line counts as an interaction with a human being 😊)
- (3) In addition to providing your code, explain your solution to each problem in the comments in your code, especially if you want partial credit.

Problem No. 1: Create a class called Lamp that has the following attributes:

- enum **STATE**: only two possible values {ON, OFF}
- static constant **ac_power**: initialized to a value of 60.0
- protected data:
 - **length**: length of the lamp
 - **height**: height of the lamp
 - **width**: width of the lamp
 - **watts**: the number of watts of the current light bulb used in the lamp
 - **brand**: the brand of the lamp
 - **state**: a variable of type STATE that is either ON or OFF
- public methods include:
 - **constructor with no arguments**: sets all internal data to “-1”
 - **destructor**: cleans up memory
 - **bool set_brand(char* arg)**: sets the variable “name” to arg.
 - **bool debug(FILE* fp)**: prints out the values of all internal data in the class; assume fp is a value file pointer that points to an open file (e.g., stdout).

Also create a main program called foo.cc (binary is “foo”), that supports the following interface:

```
foo Hirsch Tiffany
foo Hirsch Tiffany Stiffel
foo Hirsch Tiffany Stiffel ...
```

This program handles a variable number of arguments (e.g., 7 different lamp brands). It creates an array of objects Lamp and creates one object for each lamp specified from the command line. I can specify any values from the command line (e.g., don't hardcode “Tiffany”).

Once the objects are created, you will write a loop over the array that calls the debug method for each object. The only output from your program will be this debugging output showing the values of the class data. No other data should appear on the output stream.

Before the program exists, make sure memory is properly cleaned up.

You will turn in four files:

- Makefile
- foo.cc: contains only the main program
- foo.h: contains the header file defining the class
- foo_00.cc: contains the implementations for all class methods

Problem No. 2: Download example_18 from the git repository. Compile and link the program. Run the program using the command “foo vw_bug”.

Debug the program and fix the code WITHOUT changing the main program or any of the arguments to the function calls. If properly debugged, the output should be “hello world: vw_bug”.

There are a number of things wrong with this program. Beware, just because you get it to run, it still might not be correct.