Alicia Prather

Assignment 6 comparisons:

Case 1: Alicia Prather vs. Benjamin Tate

Benjamin and I took a slightly different approach to this program. Besides trivial items such as notations and naming conventions, the key difference between our programs was in how we initialized private variables in the functions. He set his private variables directly, while I used the set functions for that purpose. Set functions are an important aspect of the program because it prevents the private data from being corrupted. The other difference, which was an error on my part is that I set my defaults to 1 instead of 0 as the instructions stated. Another noticeable difference between my program and all the others is that I used #include to include unnecessary elements in many of the documents. I also like that Ben's notes are succinct than my own. Another difference between my program and Benjamin's is that he did not pass the variable by reference as stated in the instructions. I had some difficulty with this section of the assignment because I was confused about what needed to be a constant. However, in the end passing a constant reference helps maintain the data so it does not become corrupted.

Case 2: Alicia Prather vs. Alexandra Permar

Alexandra's program appeared flawless as I reviewed it. Her annotations describe sufficiently what is happening in the program without being redundant or unnecessary. Mine on the other hand were at times unnecessary. As stated in the previous statement, I #included items that were unnecessary and Alexandra did not. Another difference that I noted is that in the distanceTo function, Alexandra created additional variable to handle the difference in slopes. While that did affect how the program works, I think it is unnecessary and it seems that any addition of unnecessary variables might lend the program to errors if the program were to be modified. However, overall I thought Alexandra's program was superior to mine.

Case 3: Alicia Prather vs. Yannick

Yannick's program is very simple. It contains little to no notations and no redundancies. It is clean and efficient. The lack of notations makes it easier to read because there is less to read and sort through. However, if there had been errors or if modifications needed to be made, it might be more difficult to find the source of any problems. My program on the other hand contains more than the necessary quantity of notes, but if the program were to be modified, it would be easier to sort through the details of the program. As I mentioned about Alexandra's program, I did not like how her program created additional variables in the distanceTo function. Yannick's program was similar to mine in that function, and I think that is great advantage.

Improvements:

To improve my program, I would take some aspects from each of these.

1) I would be careful to avoid including files that are not necessary.
2) I would follow directions more carefully to be sure to initialize variable properly.

3) I carefully place notations within the program with purpose. I want my program to have clear notes that are easy to follow. I want to avoid having notes that obvious or don't serve a purpose.
4) Naming conventions. I would like to take more time considering how I will name variables. If variables have names that make sense it will be easier to follow my own program and easier for others. One thing I noticed as I was reading the other programs is that I had to often figure out the purpose of some of the variables. Variables with very obvious names (even if the name was longer) were easier to follow. Notations for the purpose of variables are helpful as well.