There were a few frustrating problems that did not allow the code to run correctly in the beginning:

1. **Computation errors/order of operations**: there were problems in getting the correct numbers because of the misplacement of parentheses that produced logic errors.
2. **Conceptually:** it was a difficult task to try to understand before attempting to program the calculator. This was especially prominent with the “mutual fund” or “royalty” cases; their fees had to be carried over the different intervals of fees.
3. **Condensing/simplifying:** I got confused in my own work because my if statements were so large and complex. The program began with many nested if and else statements, but I was ultimately able to eliminate unnecessary lines of code that repeated true values.

The following could be used to test the program:

1. **Asset values of 0, 1000, 10000 with any asset name or category.**

They were borderline case values that were turning points in the calculations. This testing was used to reveal whether or not there were conditional issues (wrong signs, etc.).

1. **Testing the strings “mutual fund” or “royalty” at 0, 500, 1500, 10000.**

This is used to show that the discount for these categories is actually valid in important turning points of the program.

1. **Testing empty strings for Asset Value and Category, in addition to negative values for “value (in thousands)”.**

This would show whether or not the program correctly outputs the alternative statements and terminate the program. It also shows whether or not the return values are terminating the program.