Project 1

**Analysis: Describe the problem including input and output in your own words.**

The program will have input and output files. The program will first ask the user what the name of the output file is. Assuming the input file is in one of the few acceptable formats, the program will reformat the contents of the file and output it onto the output file.

**Design: Describe the major steps for solving the problem.**

This program has very specific and separate steps in order to accomplish everything. It’s separated into inputting the file, putting everything from the file into an array, formatting the array, and then outputting the array according to how each question type should be handled.

It will first get the file from the user. Then it will input everything from the file into a 2 dimensional array. The format of the array is such that each row is reserved for each question and the columns are each individual line for its corresponding question. Before inserting them into the array, each line is sent to the parse function from the Parser Class. If there are answers at listed at the end of the of the input file then it will format them at the end of the array and then send it to formatAnswersFromList(). If the answers were given with the questions, then this step is skipped.

The optional formatAnswersFromList() function finds which answers were given and then appends an asterisk . The point of this function is that so regardless of how the answers were given the array would be formatted in the same way before being sent to the next function.

When a String gets sent to the parse function it checks to see if there are more than one significant lines in one of the inputted Strings. If there is more than one then it will return an array that will input it into the original array accordingly. It will also get rid of “Type:” from the first column in the question’s array and only leave the question type. It will then

Finally, the array is sent to the output function which has very little to do. The way the array is organized after the formatting makes it very simple to find a question, its type and send the parameters to its corresponding Class Q function.

Each class Q function does things differently; each will receive its correct amount of parameters, format the parameters by adding indentions and saying whether or not the answer is correct, and then returning it to the outputHandler() so that it can write it to the file. They also adds tab delimiters and “correct” or “incorrect” if necessary. One thing they all have in common is that most of them use the functions for trimming the answer and questions. trimQ() gets rid of the numbering and blank spaces. trimA() gets rid of the blank spaces and then removes the asterisk.

**Testing: (Describe how you test this program)**

There were many parts to this project and to be more efficient, each one was tested individually.

To test the function readFile(), I outputted each valid line that was read from the input file. The output was formatted as such: “arrayName[question][i]= line”. This helped visualize each individual line as it was being entered from the input file. It made it easier to check if blank lines were being ignored or not. It also served to show that the row was increasing when it was supposed to; when a new question appeared.

To test the function outPutHandler(), I would output the String returned by the Q class functions, and the contents of the array. Since this function was very basic, there was not much else to focus on. I just had to make sure that the Q class functions were returning the Strings in the correct format, and that the contents of the array came in in the correct order.

To test the parse function first the String parameter is outputted. Then since unless there are more than one significant lines I the String the function does not do much. I would output that the String has more than one significant lines. Then output everything from inside the loops that take care of the String when it has more than one significant line.

To test the formatAnswersFromList since it’s not always used I would first output that the function has been called and output the String parameter to double check that it was called at the right moment. I would then output each variable as it comes along to make sure that it’s doing what it’s supposed to do. I also would output when more than one comma was found and the values of count and iAt, and then adjusted accordingly to get the right strings into the array.

All other smaller functions were part of the aforementioned functions but I later separated them into other functions for the sake of keeping everything in better order and make the program more concise. To check them I just would output their parameters, depending on the function, the variables in between steps, and then the variable that was to be returned. It was an easy check because if they weren’t what they expected them to be it would be a very simple fix since they were small.

Once everything was sorted out, and in order to check that the format of the output file was perfect, I did a while loop that iterated through every single one of the example input files. I also wrote a small program that would check the output files I created with the ones given to us as examples. It would output when a line from a file was different from the other file’s, and it made it very easy to check, go back, and fix the small formatting errors.