1. Did you like your program topic? Why (not)?
I disliked Bingo at first which is why I switched to Connect 4. I found Connect 4 a lot more fun than Bingo because A) the UI was much easier to implement and B) I could add all these cool features (I'm very proud of all the main menu stuff) whereas Bingo was literally just a random number generator until somebody won maaany clicks later. The overall topic was perhaps on the simpler side which meant that I didn't get to really utilize anything super cool, which is a downside, but I liked how liberal I could get with what I could add to it.

2. Did your team stick with your initial bottom-up approach to program development, or did you switch to a different design approach at some point? Which approach did you prefer and why?
Yep–we started with all the functions needed and the main code structure up from there. I always prefer bottom-up programming because it means I can extensively test the code for bugs rather than having to wait until everything's done to run into some game-breaking issue. It also means I can add more functions as I go when I have an idea in mind.

3. Did your final design closely match the original hierarchy your team developed? In what ways is it the same? In what ways is it different?
It's about the same? You can't really deviate from Connect 4's overall structure and I don't know how to code in top-down. I think the game menu deviated a bit because it required a bit more tweaking to make and since it came *after* the game was already finished, made me realize how much I disliked top-down since it required extensive debugging to shove a new function and implementation into already-done-code.

4. Briefly describe the thing(s) you learned on your own, beyond what was covered in the lectures. How did you incorporate them into your program?
Well it didn't say how advanced it had to be, just beyond the lectures. So I imported a module that we hadn't used before (playsound) and used it to make a cute "clink" noise whenever you made a move. I thought about adding background music as well but that would probably make the program crash (I did have recursion originally for the main menu but A) that was eating memory like crazy and B) it was calling itself too much and ruining everything)

5. If you had more time to work on your program, what additional features would you want to add? Is there anything about your submitted program you would change?
I was going to add a way to view the .txt file through the main menu but I decided that it was unnecessary anyways. I *really* wanted to figure out how to make a live animation of the chip falling through the board but unfortunately I didn't get the time to.

6. What was the most difficult part of this assignment? Please explain.
Debugging was making me lose my mind. The game itself was first developed and ended up being finished in around 1h30m–it was really after adding the main menu that things started to go off track and took way longer than expected. The main menu caused far too many unintended consequences and maybe next time I'll work on that first rather than trying to shove it into already predetermined code.

7. Estimate the portion of the assignment completed by each member of your team (yourself included). Please explain any significant workload imbalances and give a brief summary of who did what. For this question it's ok to make a bulleted list instead of writing complete sentences.
a. Andrew: 25% - Tied the main body of the code together as well as creating the main functions
b. Mason: 25% - Implemented the file portion of the code as well as the sound idea
c. Daniel: 25% - Designed the hierarchy of the program as well as the give_rules and give_credits functions and put comments on the code
d. Jacob: 25% - Fine tuned the code with debugging and test cases