**Completeness**

For this section, I believe I have completed majority of it. The snake game builds and functions correctly. There are some bugs and issues that could be fixed along the way. I want to eventually add a Hiscore board underneath the Score portion of the screen. I also would like to go through and change up the style and design of the actual game instead of just a solid black screen with green text.

**Appropriate Complexity**

I believe that I could have worked on something a little more complex. This class taught me a lot about object-oriented design and how to change up my coding style instead of writing all of my code in just one solid block of code. I’m not a Comp Sci major but it was nice to learn and acquire the knowledge of object-oriented design.

**Version Control**

I wasn’t sure how to work github. I finally downloaded it towards the end of the project so I don’t have much of a history of my project progress. My github link is <https://github.com/Valyrixn88/CSC260Project.git>

**UML Diagram**

I have attached my UML Diagram in my github along with the project.

**Documented Test Cases and Testing**

I started the project on July 30th , I had programmed the code and the basic structure and design. Throughtout the weekend I worked on the project and had my parents play the game as I figured they would be good test subjects. I got ideas on design, changing the size of the game, the speed of the snake, whether a circle or square looked good. After taking their ideas into consideration I returned home and put their changes to the test and came up with the final project that has been uploaded to github on the 4th of August.

**Classes and their components**

I do have 4 classes, but as mentioned in the rubric the form class does not count as one. So, all together I only have 3 classes. I feel like I could have programmed the grow, dead, and createFood class could have been put into their own class, but ultimately, I decided to take the risk and put it into the form class.

**Inheritance**

I believe I have accomplished this task by having the form class inherit information from the the gameSetting class, the Movement class, and the Snake class.

**Encapsulation**

I believe I have accomplished this task by having private methods, public static methods and private static methods that only allow the information that is needed.

**Polymorphism**

I believe that I have accomplished this task by having the snake take on the growth of the food, and still being considered Snake.

**Abstraction**

I believe I have accomplished this only allowing the gameSettings class the details of the game, while only calling the changes that need to be changed to it in the form class.

**Advanced Topic**

I feel like I have accomplished the Code Organization as well as the clean and maintainable sections of the advanced topic as in my personal experience I believe the code does look clean, it’s well organized and it’s maintainable.