# Software Design/Engineering Narrative

The artifact that I chose to work with is a student management system that I have been working on in my spare time. The basic idea behind this program is to allow the user to enroll students into the school. It asks you to enter student names, the year they are going into, and what classes they would like to enroll in. As you enroll in classes it allows you to view course assignments for that course and adds the price of the course to your overall balance. Once you have your classes set up and the balance, it will ask you if you want to pay any money toward the tuition. Then the program will print out all student objects to the screen showing you the student’s information, classes, and professors, as well as the remaining balance each student owes. I started working on this artifact a few months ago for the main purpose of practicing and improving my object-oriented programming skills in java. I created this artifact with the intent that I would continue to add to it over time and look for new ways to optimize and enhance this program.

The reason I chose to use this artifact is because it showcases all the skills I would like to outline. I wanted an artifact that was a little more complicated than a simple one-page program that accomplishes a very simple task. I feel the best way to showcase your skills is to add a little complexity to the program because it forces you work on your debugging and error handling skills as well. I utilized my software development design knowledge to organize this program in a way that would make it easier for other programmers to read and understand the intent behind my code. The areas of the design portion of this assignment that you will notice are the comments in my code, the different classes and functions to separate functionality in the program and spacing and naming conventions that were used to make the code easier to read and follow. The enhancements I made in this area, aside of naming and comments, is adding in additional functionality to the program which allows the user to view course work for each course they choose to enroll in. This was done by creating a few arrays that hold and organize the course work for each course. The arrays were used because they hold a fixed number of assignments per course which allowed me to utilize this data structure. I then added in functionality which asks the user if they would like to view the course work for a course they just enrolled into. If they user chooses yes, it outputs the work to the screen and then asks if they would like to enroll in another course. If they say no, it simply moves on to asking if they would like to enroll in another course or quit.

For this portion of the enhancements, software design/engineering, I was able to complete the enhancements that I had planned. My enhancements were to make the code more organized through the use of comments on any areas of code that may look confusing to other programmers. I want to make sure that the code is easy to follow and understand. I also did this through the use of correct naming conventions for which I utilize camel case text along with descriptive names for what each function is trying to complete. Looking at my code, you will know exactly what my intent was when writing each function just by looking at the name of that function. Aside of making sure I met design best practices, the most difficult enhancement I wanted to add in was the added functionality described above. I wanted to make it possible for students to view course work for each course which was completed successfully.

While working on these enhancements, I learn a lot about my own design and best practices to implement. I start by looking over the code, looking at the overall structure of the code and try to put myself in other programmers’ shoes. This helps me identify areas of my code that may be difficult for someone else to understand which is where I added comments and fixed up naming conventions. The enhancements force me to make sure I am utilizing coding best practices that would be best if collaborating in a team environment. Other important skills that I worked on during these enhancements are my code review and debugging skills. Debugging and security are probably the most important skills that we need as programmers and while implementing these new enhancements, I was able to improve in these areas. While I was adding in the new functionality for looking at course work, I started getting error messages for other areas of my code. This took me a while to figure out because looking at the code, it didn’t appear to have any flaws that would cause those errors. In fact, if I ran the program and typed everything as it should be done without trying to look for bugs, the program worked perfectly. However, if I typed an input that wasn’t supposed to be typed in, the error handling worked on the first try but then output an error if you did it twice. This was something I hadn’t planned for initially which is why these planned enhancements were a great idea. I found a bug in the program that I didn’t realize was there. The method I use when it comes to these types of bugs is adding in print statements through portions of the code to see if my programming is reaching specific functions and to find where the program is breaking. Through this process I was able to find where the error was and correct it. Because of the added functionality, I not only worked on my error handling and security, I repaired areas of the programming I didn’t even know needed fixing. This forced me to understand that the importance of testing and retesting my code and making sure to identify all possible bugs and security risks.