The advance programming class was the final class we took in learning C# coding language. In this class we learnt about Asynchronous Programming and created WPF applications. Asynchronous Programming or Async Programming is a type of programming where a part of works runs separately from the main application and notifies the main application about its progress whether it has been completed or failed. This type of programming is very useful because it gives the platform for an application to improve and is more responsive. We used a background worker class to execute an operation on a separate thread.

WPF stands for Windows Presentation Foundation. It is a Graphic User Interface (GUI) used along with the .Net Framework. A GUI Framework allows us to create a windows application with a range of elements such as labels, buttons, textboxes and so on. We incorporated the use of background workers that allows us to start, update and cancel an asynchronous operation. We also learnt about data centric applications and data binding in this class. We used XAML to create our WPF applications as it has a greater binding capability and also provides hardware acceleration. XAML stands for eXtensible Application Markup Language. It is a newer language and it also splits the GUI and the code behind the application into several entities, making the application efficient. We learnt how to create single-object binding and binding into XAML declared instances.

We also learnt about ADO.net (ActivexDataObjects.Net) which provides the classes to develop database applications in the .Net framework. The ADO.net uses a disconnected data architecture and the concurrency issues are not handled by the database management system. The artifact that I chose for this class is the T-shirt stock application that was our final project for this class. We used the MVVM pattern that is used in WPF applications to create this application. MVVM stands for Model-View-ViewModel, which facilitates a separation of development of the GUI from the business logic. The view model is in charge of exposing the data objects from the model in a way that the objects are easily managed and presented. It is also considered as the blue between the view and the model. MVVM is basically a separation between logic and the view code. It also leverages data binding.

I chose this particular project as my artifact because we spent about a month to finish it. We were working on it as we were learning about it in class. I considered this to be effective as I was learning by doing. Apart from using the MVVM model, we also used the background workers class, connected a database with our application and used XAML to create our windows forms. This class was a fairly difficult class. However, we learnt a lot of new concepts in this class. While doing this project, there were times when I got frustrated because I couldn’t find the solution to some errors even while debugging. We worked on a in class project similar to this serving as a base for this project, however, at times, it was hard to figure out why the code wasn’t working. If I were to change anything in this project, I would make a list of what needs to be done, sequence the list and then start coding and designing the form accordingly.