A1

This program demonstrates our ability to use many different data types and expressions. We used the string data type, double data type, as well as using a class as a data type to create a variable that can access that class. We demonstrated our ability to conduct arithmetic with integers because in the FXMLDocumentController.java file, we made our program able to add, subtract, divide, and multiply, integers for the calculator to function. We demonstrate the understanding of type conversion because in the FXMLDocumentController.java, I had to convert strings to doubles in order for the calculator to conduct arithmetic with the user input through buttons. I demonstrate the ability to use non-numeric comps because our program required two strings to be compared with each other. I demonstrate the limitations of finite data representations by rounding the numbers to the maximum amount of digits that could fit into a calculator screen in the program.

A2

The program demonstrates a modular program that is divided into multiple files because considering our main focus was on working with JavaFX, we had to import many libraries and create different types of files such as .java, .css and .FXML files. The program used modular design concepts to support reusable code by implementing overriding and organizing code into methods. We demonstrated the ability to modify existing modular program code by allowing different methods to control one particular part of the GUI. For example, there were 4 types of buttons and each method was responsible for a certain type of button. We also created different files that edit the properties of other files. For example, the calculator.css file changes some properties of the FXMLDocument.fxml file to make the calculator look better.

A3

We demonstrated the ability to create algorithms that read from external files because our program had to read and write to an external XML file in order to show the GUI. Our program shows our ability to design algorithms because our calculator had to perform many different tasks with the values the user inputted through the GUI.

A4

I demonstrated I could work independently because considering I had to learn how to use XML to create GUI from nothing, I had to make use of tutorials and manuals to learn a whole new language inorder to create GUI all within a month, which is a very impressive feat. We aslo had a testing plan as we tested our program whenever we reached a certain milestone. We also fully documented our program code according to java doc standards and our documentation was exceptionally clear and organized.

B1 I demonstrated the ability to manage the software process effectively through all of its stages. Before I started coding this project, I created a layout of what we want the GUI to look like, and what features the calculators should have. I also made sure there was also room to edit the features and objectives of the calculator. I developed a project plan that provided a roadmap to which parts of the project will be completed first and how I will ensure that existing classes could be modified in case we change any of our plans. I also provided a clear framework that the project could build on by creating the uml and detailing what each class and method will do and how they will interact with each other. I produced software according to the specification I layed out on the proposal and made sure each feature I wrote about was coded in, although we had to make many adjustments to the original plan, which is all part of the agile software development process. I used project management tools such as a calendar where I placed different objectives that are to be achieved at different times, so that we would be able to map out our development process, although there were nemouse delays and we often had to change our timelines. We then closed the project once we compared what our program achieved to our proposal. For example, we closed our project once the calculator met all the basic expectations of being able to perform arithmetic. We also decided to close the project by confirming that the GUI of the calculator matched the initial concept of the calculator we laid out at the beginning of the development project. I often created timestamps of our code and kept saved versions so that we could always revert back to a recent version just in case the edits we make destroy the current version of the program.

B2

I contributed as a team member by planning out how we will develop the calculator and by creating the uml that me and my partner should follow when creating the calculator. I learned how to use XML and connect it with java on time so that we were able to code our project in time. I met the deadlines of completion for all the basic functions of the calculator and I was able to refine the GUI so that it looks high quality all within the required time frame. The progress during the project review was great as we all did our part and overcame the various challenges we had to face.

C1

Through this program, I demonstrated the ability to apply modular design concepts because the program made effective use of different classes and methods that work together to achieve the various objectives of the program. For example, the class FXMLDocumentControll, is built upon the GUI of the class FXMLDocument. As the name suggests, the FXMLDocumentController acts as the control panel for the FXMLDocument. The FXMLDocument creates the GUI, while the FXMLDocumentnController provide the GUI with functionality and allows the user to interact with the calculator GUI. Another example of modular design is the use of the css file to further enhance the XML graphics of the class FXMLDocument. I demonstrated my ability to integrate various different programming languages into a single program and insured that they all worked together to make the GUI of the calculator look better and easy to use for the user.

C2

I made effective use of algorithms to ensure that the program would be effectively able to classify which parts of the GUI the user interacted with, so that the program would provide accurate output to the user. For example, each type of button had its own methods which would get information about the button from FXMLDocument.fxml. The FXMLDocumentController then determined which methods needed to be used for the button and then the algorithms inside the method classified the properties of the button that was clicked. This way, each button did not have to have its own method. Instead we were able to divide all the buttons of the calculator into four methods within the FXMLDocumentController, based on the similar attributes of each button of the XML GUI. Our program also used algorithms to conduct the basic arithmetic functions of the calculator. For example, the handlerGeneralAction method of the calculator conducted arithmetic with the numbers the user inputted in the calculator.

4.
Based on the curricular ties noted above, our program should be assessed based on the GUI we created using Java fx, and how we made sure that the user would be able to effectively interact with the GUI. Considering this was our first time using JavaFX and CSS, it was very difficult for us to combine all the different classes and programming languages together and to get them to work with each other. We also had to learn about the many libraries that had to be imported and what each of the libraries does and how to work with them. The learning process was very difficult for us and we had to dedicate countless hours towards learning these new programming languages and how to implement them within the project. Aside from coding the GUI itself, ensuring that the user would be able to interact with the program was very challenging as well. We had to connect the FXMLDocumentController.java class, which was the java file that controlled the user interactions with the GUI, to the FXMLDocument.fxml class, which contained all the XML code for the graphics and the calculator. We also had to find out how to use CSS to further style the XML code with different colors and fonts. As you can see from our program, we placed heavy emphasis on the GUI of the program and allowing the user to interact with said GUI. In conclusion, I believe when marking, you should focus on our use of XML and css for graphics, and how we used java to allow for user interactions with the GUI, rather than marking based on how you usually do with other groups that did not focus on JAVAFX.

5.
I think the progress of the project was extraordinary and amazing, even though at times it was an emotional rollercoaster. Although we were not able to achieve all of our objective we first laid out in our proposal, i am never the less impressed with the fact we were able to achieve the basics of the calculator, taking into account the fact that we had to learn a whole new programming language and be able to implement it in our code. As I have stated earlier, we worked with many different libraries and concepts we never touched before, and we had to familiarize ourselves with the integration of different programming languages within a single java program, which takes countless hours and learning through trial and error. We had to overcome the challenges of making sure that the XML code in FXMLDocument.fxml was able to work with the java code in the FXMLDOcument controller to ensure that the user is able to interact with the GUI of the calculator. I also learned how to implement CSS into XML in order to style it better. The process of learning the different languages function and work together was an amazing feat that I had to go through inorder to provide the best GUI and functionality I can for this project. Me and my partner made tremendous progress with the project and our learning experience made us much better coders than we already were.

6.
The biggest struggle we encountered when planning and developing this project was facing the uncertainty of being able to learn how to use JavaFX. We took a major risk by choosing to learn a whole new language(XML), as there was no guarantee that we would be able to become proficient enough with XML to be able to create a calculator that the user could interact with. Choosing to learn a whole new programming language and java framework was a tremendous challenge and risk that me and ryan decided to take inorder to demonstrate our ability to adapt to difficult circumstances and to make the best out of the limited time and resources available to us. It also shows our sheer determination to learn new concepts and to apply them in the biggest project we have ever done. We committed ourselves towards spending countless hours and sleepless nights learning everything we can about java fx through reading documentation and watching tutorials on JavaFX. Another major challenge was making sure that we all stayed on the same page regarding what we wanted our project to be. We both had different ideas regarding how we should code our project and weather or whether we should use CSS to further style our GUI. Despite our differences, we were able to resolve our disputes and negotiate with each other so that we would always follow a single ultimate vision.

7.
If I had more time, I would probably be able to add more functionality that I originally planned for the calculator. Instead of solely basing our project on the implementation of Java fx and ensuring that the user could interact with GUI, I would have created more different java classes and dive deeper into the use of object oriented programming and put in place more algorithms for different purposes. However, I have a feeling that I would have instead procrastinated more and not really get much more done, even though me and my partner are very hard working people with a burning desire to learn as much as we can.

8.
Agile development process helped with developing our project because it allowed us to change course whenever we felt like we had to and it allowed us to view our progress being made and what will and will not be feasible before we move on to the next sprint.By diving our the development process into sprints, we were able to test our code and see weather or not the current plan we are following is the most efficient route to take. It allowed us to revert back to old versions of our code if we made a mistake, as part of the agile development process to save old versions of code that we could always fall back to. The agile development process gave us a lot of flexibility in terms of navigating how we will develop our code and making necessary adjustments to our starting plan. We were able to view our project through an ever changing viewpoint and considering that we were learning XML while programming, being adjust our course of action based on the new information we recently learned was of utmost importance to us, and only through the agile development process were we able to learn while coding and act as flexibly as we did.