For my create, I developed the algorithms to add functionality to a Tkinter calculator. First I created the parent function that calls the main parent window and then created children functions that give the interface elements inside the window functionality. Next I created two more child functions that calls the calculator and converter windows and then added child functions to give all the buttons inside functionality. For my math and logic element I had each button within the calculator concatenate or in other words adds whatever that button represents to a main string that is the problem. For logic I made it so that whenever you have a problem and click the equals button the problem is put through and if statement that checks for a decimal in the problem and adds one before solving so decimals can be displayed.

I did my project with a partner. Our first step was establishing what we wanted to do. I knew that I wanted us to do something using Python with Tkinter so the best option was a calculator. We then decided that I would do the functionality of the calculator and my partner would make the GUI. After this we both watched a video series that was tutorials for Tkinter. Once we both had a good understanding of how Tkinter worked, we decided how the Calculator would look and what functions it would have. Now we started developing the calculator and I would make functions for buttons as my partner made them and then we would combine our code and repeat.

My first problem was figuring out how to have all the buttons of the calculator be able to create a equation to be solved. My first idea was to have a variable that the numbers you click are added too then once you click a operator it sets a variable for that operator to true then anymore numbers you click are added two a second variable. The problem with this method is that the equation you could make would be extremely limited and it would also be hard to implement other functions. My solution was to have whatever button you press add what it represents to a string. I then found out python has a built in function called eval() which allows you to do math that is in a string. This allows the calculator to have an indefinite string and solve it.

My second problem was that whenever you did division like 5 / 11 it would display 2 instead of 2 and whatever decimal it had. I also wanted to add a button to insert decimals. To fix this issue I added the decimal button then made it so whenever you use a decimal it makes a variable called decimal True. So when you click equals it will do the math normally and give you your answer. But if do don't a decimal then it adds a decimal to the end of the last number so that the answer becomes a float and displays decimals if any.

This code segment is an algorithm because it is a list of instructions to complete when each function is ran. The selected code segment uses an algorithm that uses a mathematical concept because all of the functions that are included concatenates aka adds a string to another string. Logical concepts are also demonstrated at the bottom of the screen shot with a simple if statement that handles decimals in the problem. The selected code has a function algorithm at the bottom with two child logic algorithms. The first if statement is a logic algorithm that checks for if the variable *decimal* is True. If it is then it runs the child algorithms under it. The selected code is an abstraction because it prevents you from having to manually type your problem into the console. This abstraction manages the complexity of the program because all you have to do is click a few buttons to get the same result of manually typing into the console.