Criterion C: Development

1. Displaying Assignments
2. Data Storage
3. Taking User Input
4. Ordering Assignments
5. Marking Assignments as Overdue

1. Initially, I wanted to display each assignment as a button, each displaying the assignment information and allowing the user to modify the assignment when they clicked on it. One issue was that there was a limited number of assignments that I could display on the screen. However, while researching, I came across JTable and JScrollPane, which allowed the user to scroll down and continue viewing the JTable, effectively allowing the user to create an infinite number of assignments. Even after implementing the JTable, there was the problem that the JTable uses Arrays, which are difficult to alter. As a result, I created a method that would make a new 2D Array every time an assignment was added, deleted, or modified, which updated the JTable as well.
2. In the program, I chose to store the homework assignments in an ArrayList, because the size would always be changing, instead of an Array. One of the problems with choosing to use an ArrayList is the use of the JTable. A JTable constructor requires a two dimensional Array, containing the data to be displayed, and a one dimensional Array containing the column names. I could not create a JTable with the ArrayList. Instead, I decided to create a two dimensional Array, and transfer all of the data from the ArrayList into the 2D Array.. In the 2D Array, each homework assignment was assigned a row for its data. I then created a JTable with the 2D Array and an Array containing column names.
3. To create assignments, delete assignments, and modify assignments, the program takes user input to perform user functions. There was a problem with using JTextFields, because the user could input anything, and there would have to be a lot of code to filter out unwanted cases. While searching for ways to list choices, I came across JComboBox, which enabled me to only give the user select options for the assignment name, date, and status. This helped me avoid having to write a large quantity of code.
4. For the application to display assignments in order, the due dates would have to be compared for each assignment. To do this, I created a method to compare two assignments and determine which was due earlier. However, this would lead to a large amount of comparisons, which I wanted to avoid. To solve this problem, with the exception of the first homework assignment, I compared the new assignment to the existing assignments, and determined the correct placement of the new assignment, without having to worry about the other assignments.

5. For checking if assignments were overdue, I used Java's Date and SimpleDateFormat class. I created a method that compared the current date to the assignment's due date, and set the homework assignment's status as overdue if the current date was "greater" (later) than the assignment's due date.