Select **two** of the following topics on JavaFX and discuss their features and provide examples for your initial post, along with your code. Include in your code documentation an explanation of how the code functions. You should also include the resources you used.

* JavaFX Line
* JavaFX Rectangle Ellipse
* JavaFX Path
* JavaFX Circle Arc
* JavaFX Polygon Polyline
* JavaFX Curve
* JavaFX Color
* JavaFX Gradient Color
* JavaFX Text

As we progress and learn more about JavaFX, new features are discovered that help create richer GUI experiences. For this discussion post, my focus is JavaFX Line and JavaFX Text.

As one may guess, the line feature within JavaFX creates a line. To make a line within JavaFX, the scene.shape package must be imported into the code, and it must be under the line class (Tutorials Point, 2024). A line can be placed within a node to display it in a program. Assigning values to the x and y coordinates for a line's start and end points determines the line's size (Tutorials Point, 2024). To use the line class and other classes in JavaFX, it must extend from the Application class.

In Java, when utilizing JavaFX Text, text is created that can then be displayed within a scene. Similar to when creating the line in JavaFX, the text class must extend from the Application class. The JavaFX text feature allows the font to be altered to a different style, weight, posture, and position in the stage (Tutorials Point, n.d.). Just like on Microsoft Word or Google Docs, JavaFX offers the option to strikethrough text by setting the Strikethrough Boolean as true. If you do not want the text to strikethrough, then there is no need to include this or set it to false since the default is false. If you do not want to set a title for the stage, you do not have to, but you can see the difference between my two codes of what it looks like with and without a stage title.

**References**

Tutorials Point. (n.d.). *JavaFX - Text*. Www.tutorialspoint.com. Retrieved March 18, 2025, from https://www.tutorialspoint.com/javafx/javafx\_text.htm

Tutorials Point. (2024). *JavaFX - Drawing a Line*. Tutorialspoint.com. https://www.tutorialspoint.com/javafx/javafx\_drawing\_line.htm

**Assignment Requirements and Grading:**

* An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
* For the initial post to be considered substantive, it should be at least 250 words in length and fully cover the topics being presented. Single-sentence definitions or responses will not be awarded points.
* Submit your post by clicking on the assignment link above, then Create Thread. You must create a thread in order to view your peers' posts. Tip: Create your post in a Word document and then copy and paste your work into the thread.
* A minimum of three (3) responses, to the original threads of other students, of 100-200 words each are due by **Sunday, 11:59 p.m., CST**.
* To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hi, Nardos! After reading your post for this week, I think you did a great job explaining and demonstrating how the JavaFX color class can be utilized. You are correct that JavaFX has many elements that create and hold all media types. I have been enjoying these added elements of JavaFX because it allows us to personalize a program as much or as little as we want. However, creating a more appealing program to use and work with helps retain users. As you mentioned, no one likes looking at a dull, colorless website. I always love it when an application gives personalization options for the users, too.

Hey, Jessica! Your included code for the JavaFX color class and Gradient color are great examples of utilizing those classes. I really enjoyed working with this in our M1 program for this week. Although I only included minimal usage of it, it was still exciting to be able to customize the program a little more and really capitalize on Java’s capabilities. I find it helpful that IDE’s can determine if a class is not imported so it makes it hard to miss those when programming. I did not really use gradients much this week, but I am excited to utilize them in the future and when it seems more fitting or engaging than a solid color.

Hello, Truman! I really enjoyed reading your discussion post, and the lines of code you included do a good job of further elaborating on your points. Although the JavaFX line class is simple, I think it is very handy, and I see how it will be beneficial as we continue on our software development journey. It is always important to remember to set a stage and scene for the JavaFX classes like line so that they create a pop-up. It feels so rewarding to watch a program properly display after running it. This is especially true when I keep running into errors along the way.