The following topics cover Advanced JavaFX and FXML. Choose **one topic from Group 1** and **the topic from Group 2** to discuss for your initial post. Provide a code example where necessary to elaborate your thoughts.

**Group 1**

* Rotations
* Menus
* SplitPane
* TableView
* Controller
* Handling Controller Events

**Group 2**

* FXML Program Development

Earlier, we learned about JavaFX, but the extent it can go is even more advanced. One of the components of JavaFX is the menu. Menus allow a user to select different options that are programmed into the menu. When a menu item is selected, it then opens the option that the user clicked, and the menu closes. JavaFX allows menus to be placed on any part of the screen. When creating an empty menu, menu() is used. Otherwise menu(String s) holds a string, menu(String s, Node n) shows a set text and image for a menu item, and menu(String s, Node n, MenuItem… i) does that same as menu(String s, Node n) but also “inserts the given items into the items list” (GeeksforGeeks, 2018). You also add submenus to a menu.

Here is an example of how the menu bar can be implemented in Java (GeeksforGeeks, 2018):

**public** **class** MenuBar\_1 **extends** Application {

    // launch the application

**public** **void** start(Stage s)

    {

        // set title for the stage

        s.setTitle("creating MenuBar");

        // create a menu

        Menu m = **new** Menu("Menu");

        // create menuitems

        MenuItem m1 = **new** MenuItem("menu item 1");

        MenuItem m2 = **new** MenuItem("menu item 2");

        MenuItem m3 = **new** MenuItem("menu item 3");

        // add menu items to menu

        m.getItems().add(m1);

        m.getItems().add(m2);

        m.getItems().add(m3);

        // create a menubar

        MenuBar mb = **new** MenuBar();

        // add menu to menubar

        mb.getMenus().add(m);

        // create a VBox

        VBox vb = **new** VBox(mb);

        // create a scene

        Scene sc = **new** Scene(vb, 500, 300);

        // set the scene

        s.setScene(sc);

        s.show();

    }

**public** **static** **void** main(String args[])

    {

        // launch the application

        launch(args);

    }

}

From group 2, FXML program development will be discussed. FXML can help build a user interface that stands alone instead of being combined with the code logic of a program (Oracle, 2025). This allows code to be more easily managed. FXML can be utilized for all user interfaces like static layouts, but according to Oracle (2025), it is especially good for handling “large, complex scene graphs, forms, data entry, or complex animation”. FXML is beneficial for creating and maintaining testable UI, does not require compliance, and is not limited to the MVC interface (Oracle, 2025).

Here is an example of an FXML markup by Oracle (2025):

***Example 1-2 FXML Markup for a User Interface***

<BorderPane>

<top>

<Label text="Page Title"/>

</top>

<center>

<Label text="Some data here"/>

</center>

</BorderPane>

**References**

GeeksforGeeks. (2018, July 8). *JavaFX | MenuBar and Menu*. GeeksforGeeks. https://www.geeksforgeeks.org/javafx-menubar-and-menu/

Oracle. (2025). *1 Why Use FXML (Release 8)*. Oracle.com. https://docs.oracle.com/javase/8/javafx/fxml-tutorial/why\_use\_fxml.htm

**Assignment Requirements and Grading:**

* 1. An initial post of approximately 250 words is due by **Thursday, 11:59 p.m., CST**.
  2. 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.
  3. 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.
  4. 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**.
  5. To view the rubric grading criteria, click on the following link: [Discussion Board Grading Rubric.](https://content.bellevue.edu/cst/csd/rubricdbv3.pdf)

Hi, Lea! After reading your post for this module, I think you did an excellent job explaining menus and working with FXML with Java. I agree that menus are important in GUI applications. Besides being disorganized and confused, I am so used to interacting with menus that I cannot imagine what it would be like without them. In addition to menu bars, there is also the capability of adding submenus within a menu. There are even options to create empty menus. I would mention that FXML allows developers and designers to work independently. This is a huge positive of FXML.

Hey, Anton. I enjoyed reading your post for this week and thought it was very insightful on the workings of TableView and FXML. I imagine that using a TableView for a program would be greatly helpful in many scenarios. I use spreadsheets so often that including them in software for business could be a great way of implementing them in real life. The code example does a great job of expanding on the description of TableView and provides a way to implement it in a program. I like how you were able to connect TableView to FXML. FXML can really bring a program to life.

Hey, Joe. Your post for this module was really well explained. You did an excellent job discussing exactly how TableView and FXML work and how they can function together in tandem. Outside of TableView, FXML can add intrigue to an application if the design aspect is appealing to a user. This makes users more likely to return to a site in the future. It also makes you more likely to stand out compared to other competitors. A good program can also appeal to new users based on positive experiences from other users. I like working with HTML and CSS, so FXML seems quite enjoyable for me.