**Module 11.2 Assignment – JavaFX Basics**

Dan Zhu

Software Department, Bellevue University

CSD402-O307 Java for Programmers (2255-DD)

Professor David Ostrowski

May 11, 2025

**Module 11.2 Assignment – JavaFX Basics**

JavaFX is an open-source framework that is built on Java. It is compatible with various operating systems like Windows, macOS, and Linux. It is a client application platform for desktop, mobile, and embedded systems on Java. It is used to create Java applications and offers a wide range of different UI components. It includes buttons, tables, menus, and, most importantly, a customized style. There are three frameworks to build UI in Java. They are Java AWT, Swing, and JavaFX. JavaFX is the only one which supports 2D and 3D charts and animation. It is designed to replace Swing. Before JavaFX, it was tedious to add media to the UI. However, so far it only supported limited Java version. It is a great choice to use to build a more modern user interface. JavaFX HBox and JavaFX GridPane are layout components of JavaFX.

JavaFX HBox is a layout component that displays all child components horizontally. You can get this component by importing class from "javafx.scene.layout.HBox". For an HBox to be visible, it has to be added to the scene graph. The HBox class provides several methods.

The HBox class has four constructors:

1. HBox(): It will create a horizontal layout without spacing and be aligned from the top left.
2. HBox(double spacing): It will create a horizontal layout with spacing based on the value.
3. HBox(double spacing, Node... children): It will create a horizontal layout with spacing between children.
4. HBox(Node... children): It will create a horizontal layout without spacing between children.

HBox has 24 built-in methods in the class and many different methods inherent from other classes. getChildren() method from the class javafx.scene.layout.Pane and backgroundProperty () from class javafx.scene.layout.Region. You can check out more methods from this link: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/HBox.html>. Here are some standard methods:

1. getAlignment(): It will return the alignment value.
2. getMargin(Node child): It will return the margin value.
3. getSpacing(): It will return the spacing between children.
4. setAlignment(Pos value): It will set the alignment.
5. setMargin(Node child, Insets value):  It can set the child's margin.
6. setSpacing(double value): It will set the spacing between children.

When do you want to use HBox? You should use HBox whenever you want to display children horizontally. You can also use different methods that HBox offers to style the children. HBox makes the display of children inline and simple.

JavaFX GridPane is a layout component that can lay components into a grid layout. A grid layout will organize contents into rows and columns. In a grid layout, all components in the same row will have the same height, and all components in the same column will have the same width. You can specify the spacing between rows and columns. By default, the number of child components controls the grid layout's number of rows and columns. However, if you insert a component into the GridPane, you can specify where to insert the component. The Grid Pane is one of the best options when you create a form. You can import the JavaFX GridPane from the class javafx.scene.layout.GridPane.

Here is the syntax of how to create a GridPane: GridPane gp = new GridPane().

There are a lot of different methods that are built inside the GridPane class and other practical methods inherited from the parent class. Here is the link about various methods that you can reference: <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html>. Here are some standard methods that allow you to add components to GridPane and add spacing between rows and columns:

1. add(Node child, int columnIndex, int rowIndex): You can add a child to a specific row and column.
2. add Node child, int columnIndex, int rowIndex, int colspan, int rowspan): It allows you to add a child in a specific row and column with row and column span.
3. setHgap(double value): It will set the gap between columns.
4. setVgap(double value): It will set the gap between rows.

To see GridPane, you must add it to the scene graph. You could use it to create a form and a table. It offers multiple methods by which you can customize the grid layout.

**References**

Jenkov, J. (n.d.). *JavaFX HBox*. Jenkov.com. Retrieved May 6, 2025, from <https://jenkov.com/tutorials/javafx/hbox.html>

Jenkov, J. (n.d.). *JavaFX GridPane*. Jenkov.com. Retrieved May 6, 2025, from <https://jenkov.com/tutorials/javafx/gridpane.html>

Naukri Code 360. (n.d.). *JavaFX HBox*. Naukri.com. Retrieved May 6, 2025, from <https://www.naukri.com/code360/library/javafx-hbox>

Oracle. (n.d.). *Class GridPane (JavaFX 8)*. Oracle Documentation. Retrieved May 6, 2025, from <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/layout/GridPane.html>

‌