Sources:

CISC 190 Java Programming

<https://www.tutorialspoint.com/javafx/javafx_installation_using_eclipse.htm>

**+Contributors:**

Trieu Huu Phuong

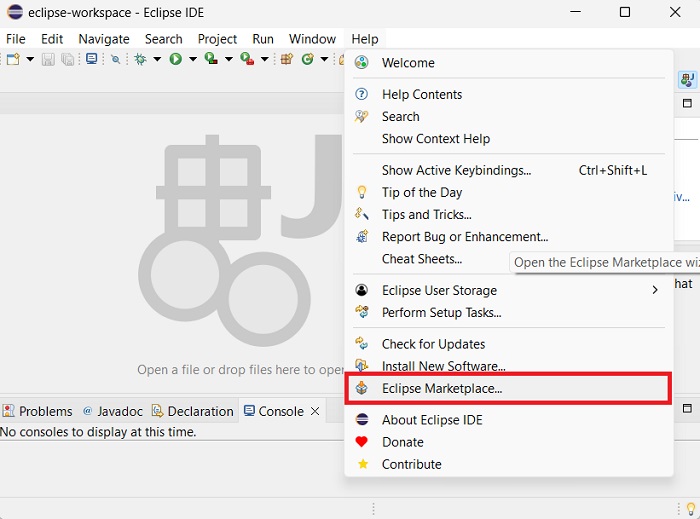
**JavaFX - Installation Using Eclipse**

**Notice that this document assumed that you already installed javaFX folder into your Java folder**

**Installing JavaFX Plug-In**

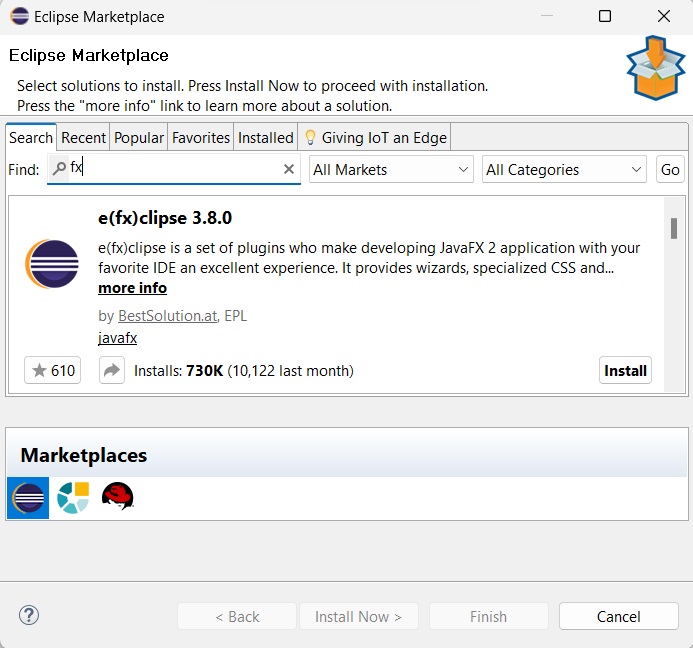
**Once Eclipse is installed, follow the steps given below to install e(fx)clipse plug-in in your system.**

**Step 1 − Open the Help menu in Eclipse and select Eclipse Marketplace option as shown below.**

****

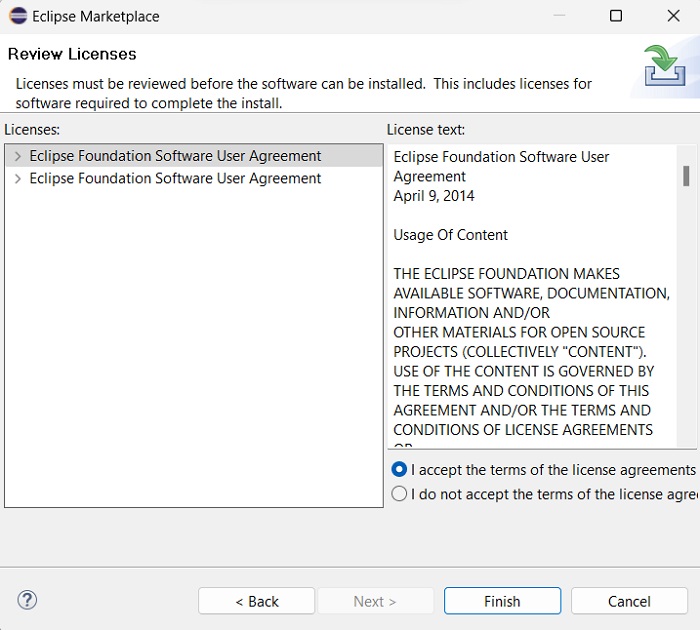
**Upon clicking, it will display the new window containing all the plug-ins, as shown in the following screenshot.**

**Step 2 − In the Find text box, search for e(fx)clipse plug-in using the 'fx' keyword.**

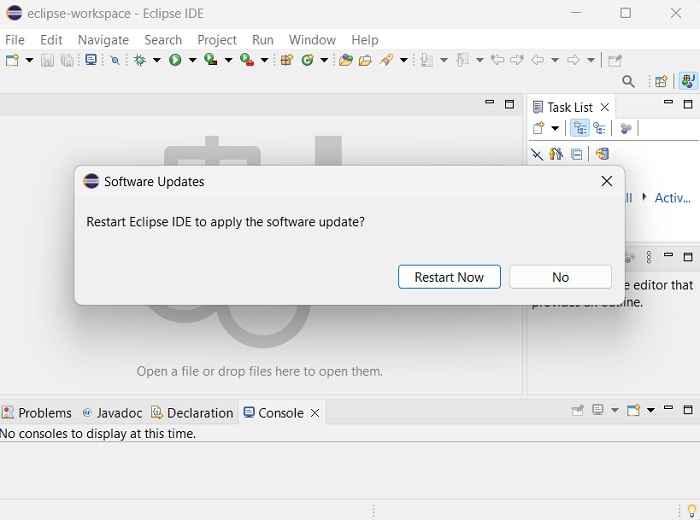
****

**Step 3 − Once retrieved, click the Install button to install JavaFX plug-in in Eclipse.**

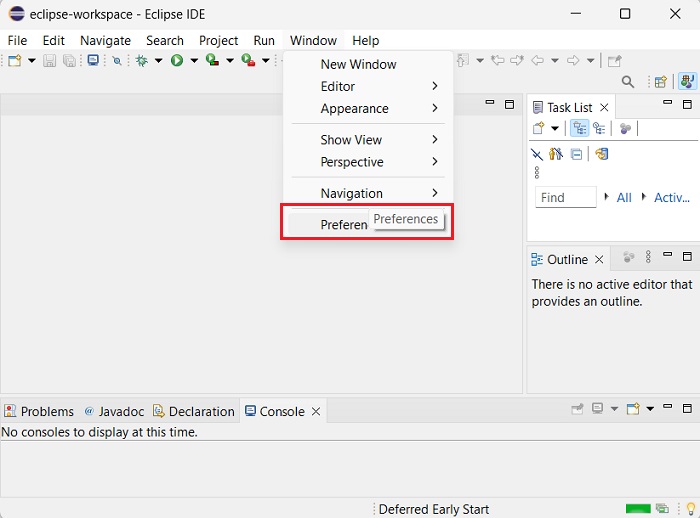
**Step 4 − In the next window, you will see a license agreement for JavaFX that you have to agree to. Select "I agree" and click "finish".**

****

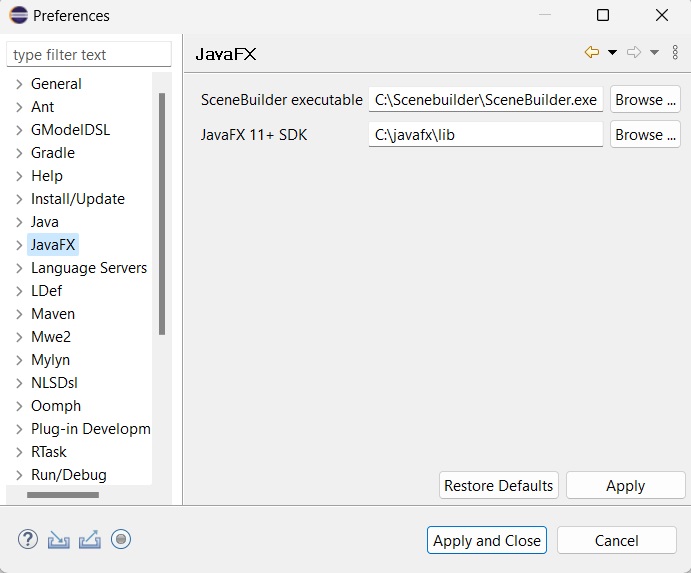
**Step 5 − Once the plug-in is installed, the eclipse will need to be restarted. So, click the restart option as shown below.**

****

**Step 6 − You will have to set the preferences by pointing to the installation directory of JavaFX. For that, in the "Window" menu, click on "Preferences" as shown below.**

****

**Step 7 − Click on "JavaFX" option and point to the "lib" directory of JavaFX SDK as shown in the screenshot. If you have "Scenebuilder" installed, you can point to the installation directory of Scenebuilder as well, like shown.**

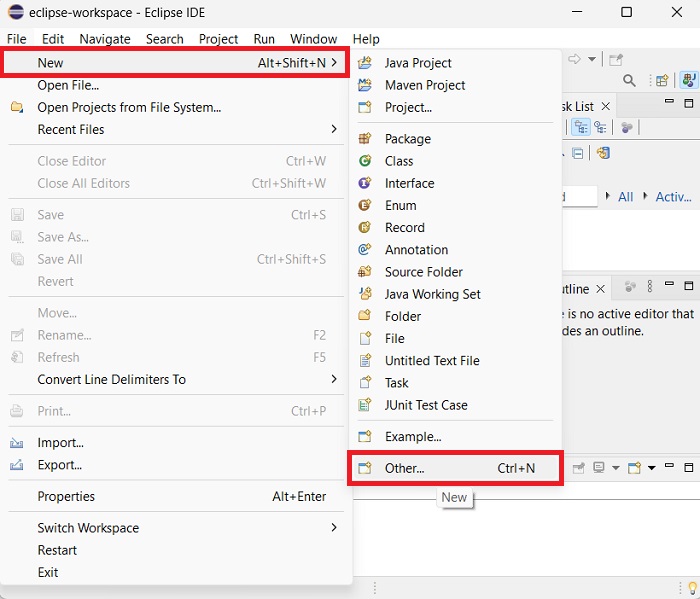
****

**=>**

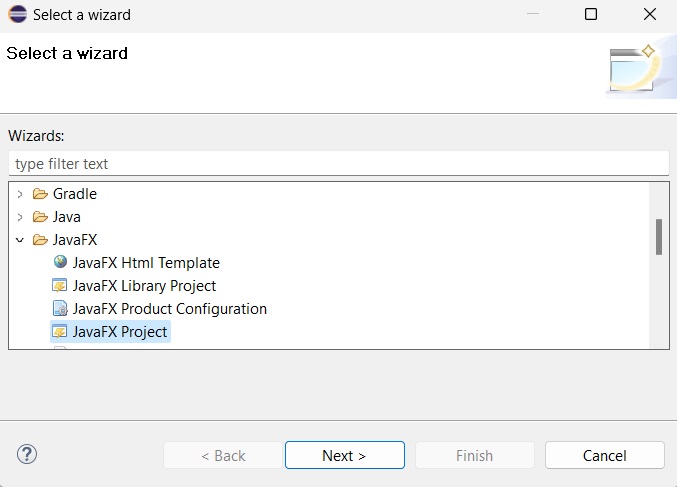
**A screenshot of a computer

AI-generated content may be incorrect.**

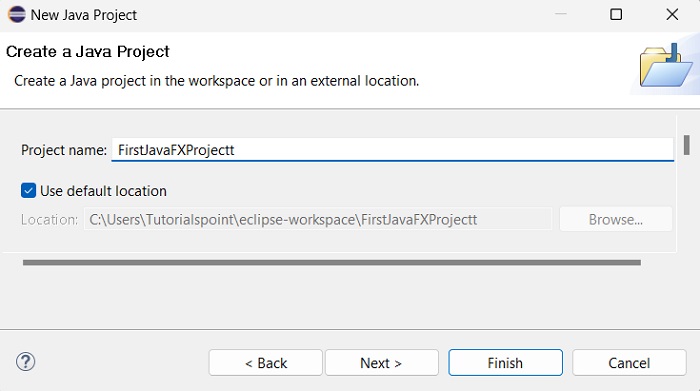
**Step 8 − Now that JavaFX preferences are set, you are ready to create a new JavaFX project. Go to "File -> New -> Other".**

****

**Step 9 − Then, you will get a window where you can see a list of wizards provided by Eclipse to create a project. Expand the JavaFX wizard, select JavaFX Project and click the Next button as shown in the following screenshot.**

****

**Step 10 − On clicking Next, a New Project Wizard opens. Here, you can type the required project name and click Finish.**

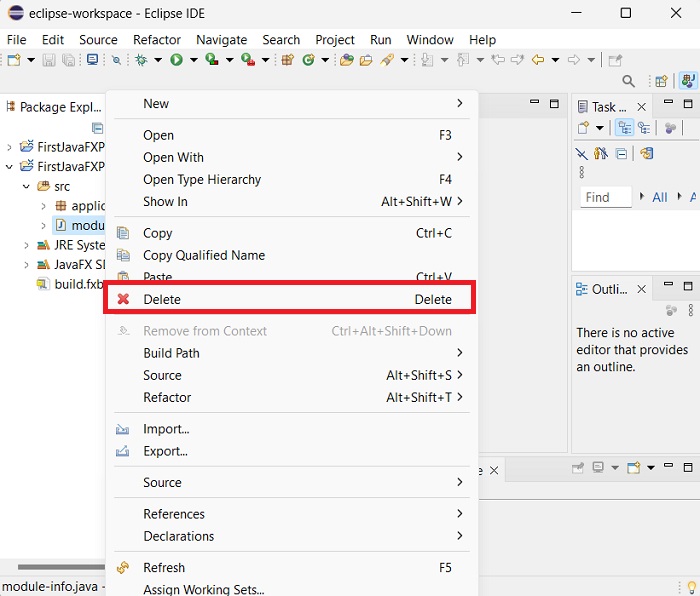
****

**=>**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step 11 − On clicking Finish, an application is created with the given name (FirstJavaFXProjectt) or (JavaFXProject). A module\_info.java file is automatically created as well, which needs to be deleted in order to avoid errors while running our project.**

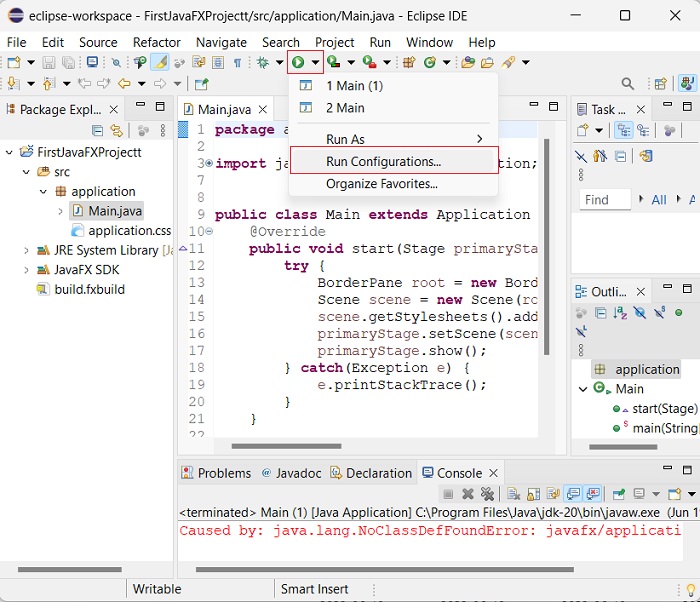
****

**=>**

**A screen shot of a computer

AI-generated content may be incorrect.**

**Step 12 − In the sub-package named application, a program with the name Main.java is created as shown below. Before executing this file, we must add VM arguments. For that, go to the "Run" symbol and click "Run configurations" as shown in the screenshot below.**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

**Turn to Arguments tab**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Step 13 − In the "Arguments" tab, add the following VM arguments −**

**In Windows**

**--module-path "\path\to\javafx-sdk-20\lib" --add-modules javafx.controls,javafx.fxml**

**for my pc:**

**--module-path "C:\Program Files\Java\javafx-sdk-23.0.2\lib" --add-modules javafx.controls,javafx.fxml**

**New version of JavaFX should be written as:**

**--module-path "C:\Program Files\Java\javafx-sdk-24\lib" --add-modules javafx.controls,javafx.fxml**

**In Linux**

**--module-path /path/to/javafx-sdk-20/lib --add-modules javafx.controls,javafx.fxml**

**Look at the screenshot below −**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Watch out for the Java Application that you want to run – In my case, Main is the javaFX application, MainClass is the Java application**

**Once added, click "Apply" and "Close".**

**Step 14 − Now, upon executing the Main.java file, a new window pops up containing the given label.**

**package application;**

**import javafx.application.Application;**

**import javafx.stage.Stage;**

**import javafx.scene.Scene;**

**import javafx.scene.layout.BorderPane;**

**public class Main extends Application {**

**@Override**

**public void start(Stage primaryStage) {**

**try {**

**BorderPane root = new BorderPane();**

**Scene scene = new Scene(root,400,400);**

**scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());**

**primaryStage.setScene(scene);**

**primaryStage.show();**

**} catch(Exception e) {**

**e.printStackTrace();**

**}**

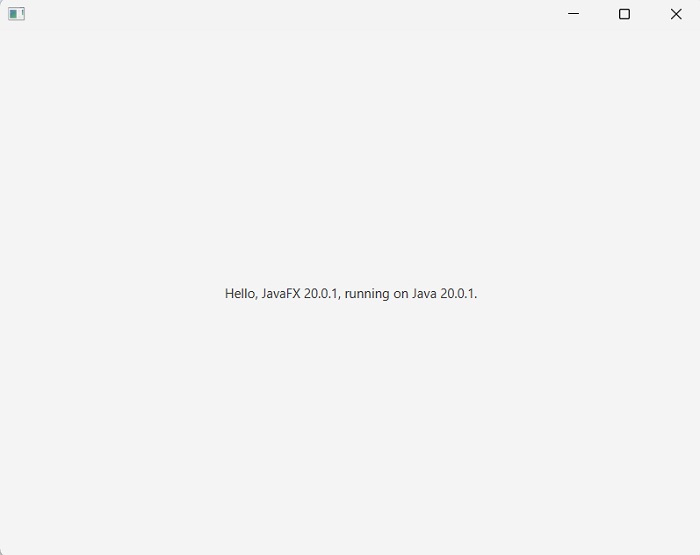
**}**

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

**launch(args);**

**}**

**}**

****

**Notice that Your Main.java file will be located in the url:**

**C:\Users\userName\eclipse-workspace\JavaFXProject\src\application**