

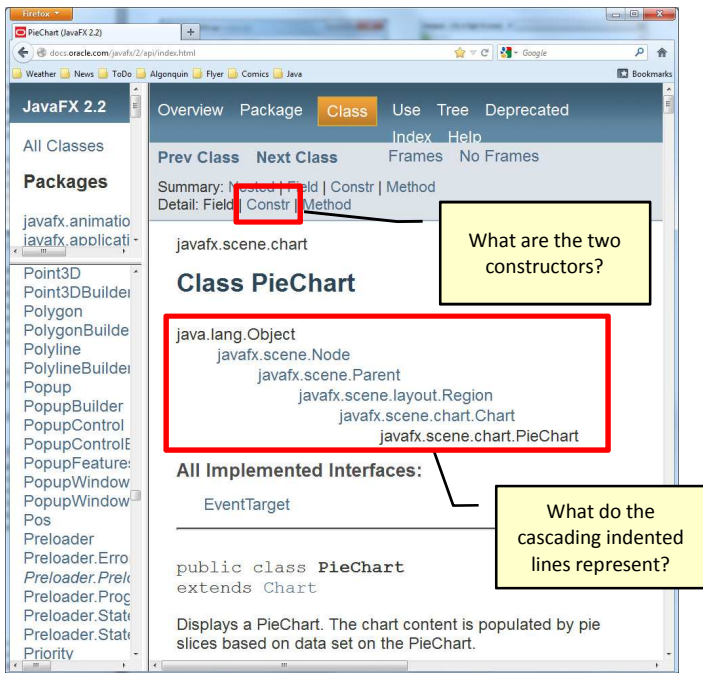
# Build Pie Chart (learning Key Terms in JavaFX)

## 1: Researching Classes in JavaFX

JavaFX provides a rich set of classes which you can use to generate sophisticated user-interfaces. You will work with several of these in the *Pie Chart* code that's part of this exercise.

In exploring JavaFX classes, you'll also need to search for material in the JavaFX API documentation. Most of you should already have the JavaFX API documentation on your local machine, but if you don't you can access the online version here: <http://docs.oracle.com/javafx/2/api/index.html> You can download the complete documentation set from here: <http://www.oracle.com/technetwork/java/javase/documentation/java-se-7-doc-download-435117.html>

Since we're working with the *PieChart* class in this exercise, you should be able to look up the class in the documentation. You'll see something like this:



What do the cascading indented lines represent?

Notice the highlighted link **Constr** It stands for *Constructor*.

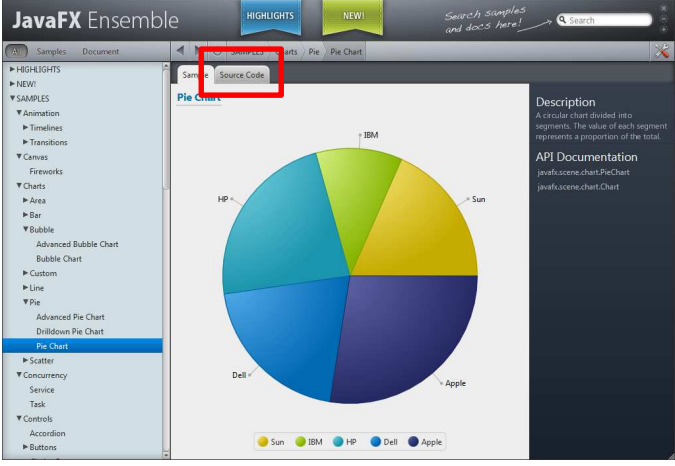
Select the index link to jump further into the *Pie Chart* documentation where the *constructors* are documented. What are the two constructors? How do they differ?

## 2: Capturing PieChart NetBeans Project

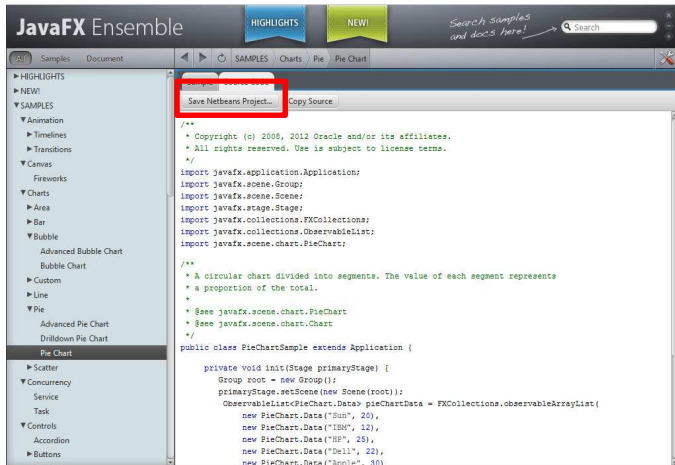
Launch the *Ensemble* sampler application, and locate the *PieChart* example.



After you select the *PieChart* example, you should see the following:



Select the *Source Code* tab and you should see the following:



Select the *Save NetBeans Project...* to capture the source code in a pre-built project. You'll need to specify the location and the name of the project (both of which you'll need to remember when opening in *NetBeans*). Your project name should have no intervening spaces or special characters (just to avoid confusing the *NetBeans IDE*).

### 3: Launch NetBeans – Open the Project

Inside NetBeans, on the projects pane, open the *PieChart* project that you just finished saving.

Run it, and the output should look much as it did in *Ensemble*.

### 4: Researching Other Classes

There are several classes used in this *PieChart* sample. In particular, there are four classes that are used with the `new` operator to instantiate objects.

Find those four classes in the source code and look up each in the JavaFX API documentation. Capture a one or two sentence description for each of the four.

One of these four classes is unusual in that it consists of two names separated with the dot operator. It is `PieChart.Data`. What does that tell you about the location of the class `Data`

Class: *PieChart.Data*

Description:

Class: \_\_\_\_\_

Description:

Class: \_\_\_\_\_

Description:

## 5: Add Two Slices of Pie

The original *PieChart* object had 5 segments. Add two more pie segments. Give these two new pie segments suitable labels and sizing

### Evaluation Guide

- PieChart* questions (1 mark)
- Class 1 description (1 mark)
- Class 2 description (1 mark)
- Class 3 description (1 mark)
- Class 4 description (1 mark)
- Modified Program Runs Correctly (1 mark)
- Good comments (3 marks). Note: Your individual comments convince me that you understand elements in the sample code.
- Source code naming, spacing and indentation consistent. (1 mark)

Total: / 10

Class: \_\_\_\_\_

Description:

### Due Date

The exercise is due by Tuesday November 13<sup>th</sup> start of lecture period 2:00 p.m.