

# Web Application Development

---

## Exercise 01

### Practicing Classes, Modules and GitHub

**STEP 1** Create a new public GitHub repo called PhotoDemo and set it up on a local folder called PhotoDemo. Share your GitHub repo URL on Slack with the class under the “week02” channel.

**STEP 2** Create a JavaScript file call photo.js that demonstrates the methods of three related classes for a company that develops photographs. Create an index.html file that loads the photo.js so you can test your app.

In the JavaScript file, create a class named Photo that includes properties for width and height in inches.

The constructor should accept values for the two properties. In the constructor, set default value for with as 8 and for height as 10.

Include a function price, which returns:

- 4 as the price for an 8-inch by 10-inch photo
- 6 as the price for a 10-inch by 12-inch photo
- 10 as the price for any other size combination

Include another function called toString which returns:

*"This is a [width] by [height] photo and it costs [price]"*

Replace the text inside [ ] with the appropriate values.

Create two objects of this class and output toString to the console.log() to demonstrate it working.

**STEP 3** Commit and push your changes to GitHub with the comment: "Create and Demonstrate Photo Class".

**STEP 4** Extend two new classes from Photo:

- MattedPhoto: This class includes a property to hold a color.
- FramedPhoto: This class includes two properties that hold the frame's material (such as silver) and style (such as modern).

Include constructors for both new classes which include not only width and height, but also the new properties of each one.

Make sure you call "super" to set the properties of the parent class.

Create a new price function for each one. The price for a MattedPhoto increases by \$10 over its base cost, and the price for a FramedPhoto increases by \$25 over its base cost.

**Tip: Remember to call the super to get the base cost.**

Each subclass should include a toString function that returns:

- MattedPhoto: "This is a [width] by [height] matted [color] photo and it costs [price]";
- FramedPhoto: "This is a [width] by [height] [material] framed photo. The style is [style] and it costs [price]";

Replace the text inside [] with the appropriate values.

Create two objects of each of the 3 classes and output each toString to the console.log() to demonstrate them working.

**STEP 5** Commit and push your changes to GitHub with the comment: "Create and Demonstrate MattedPhoto and FramedPhoto Classes".

**STEP 6** Use the module feature of JavaScript to move the demonstration code (console.log() lines) from photo.js file to another file called main.js. Use the export and import keywords to be able to do that.

Also, modify the index.html file to point to main.js instead of photo.js.

**STEP 7** Commit and push your changes to GitHub with the comment: "Convert to using module feature".