Declared Objects Lab

In this lab we're going to make our roman numeral convertor more object-oriented. We will create a JavaScript object called *number* which represents the idea of a number without worrying about how that number will be displayed. This object will have a couple of properties:

- number.romanNumeral
- number.decimalNumber

And a couple of methods:

- number.asRoman()
- number.asDecimal()

To convert from a decimal to a roman:

```
number.decimalNumber = 10;
var r = number.asRoman();
assert.areEqual(r, "X");
```

To convert from a roman numeral to a decimal:

```
number.romanNumeral = "VI";
var d = number.asDecimal();
assert.areEqual(d, 6);
```

Continue working with your pair programming partner on the roman numeral kata. Going back and forth between the partners (red, green, refactor), follow these steps.

Currently your tests may do something like this:

```
var d = 5;
var r = convertToRoman(d);
assert.areEqual(r, "V");
```

1. Change all your tests to use a format like this instead:

```
number.decimalNumber = 5;
var r = number.asRoman();
assert.areEqual(r, "V");
```

2. Run and test. All your tests will fail (red) because we've changed how to work with the function.

3. In your business logic, create a global object using a construct like this:

```
var number = {
  asRoman: // Move all your decimal-to-roman logic here
  asDecimal: // Move all your roman-to-decimal logic here
};
```

Note: number.asRoman will no longer be receiving an input parameter because it will now be using the number.decimalNumber property. Same with number.asDecimal.

4. Rerun your tests, making sure they all pass.

5. Feel free to continue the process of one partner writing a test and the other making it pass until lab time runs out.