Learning Goals

1. Be able to identify method inputs and return value type from reading how they're used in context.
2. Create and use functions/methods.
3. Follow the flow of code execution between functions/methods.
4. Reuse a helper function for shorter, simpler code.

Business Goal

You're going to provide some functions for formatting dates in a more readable format.

1. Month Day, Year (October 4, 2014)
2. MM/DD/YYYY (10/4/2014)
3. YYYY-MM-DD (2014-10-04)

You'll be writing three functions (one for each format), plus a reusable helper function to pad a number with a leading zero.

For this program, the main() will be provided for you. You have to create and fill in the matching methods. This will combine several coding techniques you've seen already, so you might find it useful to have your prior notes nearby.

Planning Your Code

1. Read through the provided source code. Figure out what it's supposed to be doing and what pieces are missing.
2. Create the missing methods, matching the names shown, and determine their input parameters and return types.
3. **Fill in the contents of each of the missing methods.**
4. **Run the code and work out the bugs so your output is exactly as shown above.**

About the Methods

`formatNumberAsTwoDigits()   [example input 3, output "03"]`

01/09/2015 requires zeros on the month number and day number, while 10/11/2015 does not. The formatNumberAsTwoDigits will take a number as input (just one number, not a whole date), and will return a String with either the number prepended with a zero (so 0-9 becomes "00" to "09") or will return the number as-is but as a String (so 10 becomes "10", 11 becomes "11", and so on). This handles just one number at a time. You can call it repeatedly (once for

Copyright © 2014 Jenny Brown

the month number, and once for the day number) and combine the results to make a pretty date. Write this method first. Understanding it will help you with the other methods.

`nameFormat()` `[example input 1, 1, 2015; output "January 1, 2015"]`

The nameFormat() takes a month number, day number, and year number, and produces a date with month name space day number comma space year number. Yes, you will need to provide code that lists out each month name so that you can convert from number to name. Notice that the day number does not use zero padding in this format.

`slashFormat()` `[example input 1, 1, 2015; output "01/01/2015"]`

The slashFormat() takes a month number, day number, and year number, and produces a date with zero padding and slashes. You'll be reusing the formatDateAsTwoDigits() method inside here so your code stays short and simple.

`dashFormat()` `[example input 1, 1, 2015; output "01-01-2015"]`

The dashFormat () takes a month number, day number, and year number, and produces a date with zero padding and slashes. You'll be reusing the formatDateAsTwoDigits() method inside here so your code stays short and simple.

Here's the expected output.

```
03
09
10
12

October 4, 2014
10/04/2014
2014-10-04

January 1, 2015
01/01/2015
2015-01-01

December 31, 2013
12/31/2013
2013-12-31
```

Copyright © 2014 Jenny Brown

Provided Starter Code

```
public class Dates
{
    // --- add methods here to make the program work ---

    // formatNumberAsTwoDigits

    // nameFormat

    // slashFormat

    // dashFormat

    // ---- end of your work area -----

    // don't change this code.
    public static void printDate(int year, int month, int day)
    {
        System.out.println(nameFormat(year, month, day));
        System.out.println(slashFormat(year, month, day));
        System.out.println(dashFormat(year, month, day));
        System.out.println();
    }

    // don't change this code.
    public static void main(String[] args)
    {
        System.out.println(formatNumberAsTwoDigits(3));
        System.out.println(formatNumberAsTwoDigits(9));
        System.out.println(formatNumberAsTwoDigits(10));
        System.out.println(formatNumberAsTwoDigits(12));

        printDate(2014, 10, 4);
        printDate(2015, 1, 1);
        printDate(2013, 12, 31);
    }
}
```

Copyright © 2014 Jenny Brown

Hints

Don't overthink this. You don't need to use anything more than you've already learned. There's more code to type, but it's very straightforward code.

Make sure you know how to create a method, and how to call a method and save its return value in a variable. Review the syntax if you need to. Think about the data type that goes in to each method, and the data type that comes out as a return value.

You can reuse the formatNumberAsTwoDigits() method from inside your date methods, so you don't have to repeat that code.

Are you uncertain what this assignment is asking you to do? Here's a complete sample program that follows the same pattern but solves a slightly different goal. Run it and examine how it works. Figure out what each step does. Then look at this assignment again.

```
public class PrettyNames
{
    // --- add methods here to make the program work ---
    public static String lastDashFirst(String firstname, String
lastname)
    {
        return lastname + " - " + firstname;
    }
    // ---- end of your work area -----

    // don't change this code.
    public static void printName(String firstname, String lastname)
    {
        String combinedName = lastDashFirst(firstname, lastname);
        System.out.println(combinedName);
        System.out.println();
    }

    // don't change this code.
    public static void main(String[] args)
    {
        printName("Jenny", "Brown");
    }

}
```

Copyright © 2014 Jenny Brown