Formative Assignment for Module #2, Lesson #3: Classes

The overall goal of this assignment is to implement a Ruby class with

- class attribute(s)
- class method(s)
- initializer method(s)
- instance attribute(s)
- instance method(s)

The functional goal of the assignment is to write a `Person` class that will maintain state thru instance and class attributes and provide access thru instance and class methods.

Functional Requirements

1. Write a `Person` class with a
   - `first_name`
   - `last_name`
2. Track each instance of `Person`.
3. Search for a `Person` by `last_name`.

Getting Started

1. Download and extract the starter set of files. The student-start directory of this starter set will be referred to as the root directory of your solution.

   ```
   --- student-start
     |-- .rspec (important hidden file)
     |-- module2_lesson3_formative.rb
     `-- spec
         |-- lesson3_spec.rb
         `-- spec_helper.rb
   ```

   - .rspec - configuration file for unit tests. If you move your files you must take care to also copy this file.
   - module2_lesson3_formative.rb - contains the starting examples. Your solution must be placed within this file.
   - spec - this directory contains tests to verify your solution. You should not modify anything in this directory
2. Install the following gems used by the rspec unit tests. You may have some of these already installed.

   ```
   $ gem install rspec
   $ gem install rspec-its
   ```

3. Run the rspec command to execute the unit tests within the spec directory. This command should be run from the root directory of the project. This should result in several failures until you complete your solution.

```
$ rspec

FFF.

Failures:

  1) lesson3 check results unexpected search result
  2) lesson3 check instance properties missing first_name
  3) lesson3 check instance properties missing last_name

Finished in 0.0184 seconds (files took 0.11245 seconds to load)
4 examples, 3 failures

Failed examples:

rspec ./spec/lesson3_spec.rb:13 # lesson3 check results unexpected search result
rspec ./spec/lesson3_spec.rb:21 # lesson3 check instance properties missing first_name
rspec ./spec/lesson3_spec.rb:25 # lesson3 check instance properties missing last_name
```

4. Implement the solution and re-test.

Technical Requirements

1. Implement all parts of this assignment within the module2_lesson3_formative.rb file in the root directory of your solution. The grader will load this specific file from this location.
2. Your script must contain a `Person` class
3. The `Person` class must have
   - a first_name and last_name attribute with public accessors for setting and getting the attributes
   - a class attribute called `people` that holds an array of objects
   - an `initialize` method to initialize each instance
   - a `to_s` method to return a formatted string of the person's name
   - a `search` method to locate all people with a matching `last_name`
4. The `Person initialize` method must
   - accept two parameters (first_name and last_name) and use them to initialize the first_name, and last_name instance attributes.
   - insert the created instance (self) into the `people` class variable
5. The `Person to_s` instance method must
   - return a formatted string as `first_name(space)last_name`
6. The `Person search` class method must
   - accept a `last_name` parameter
   - search the `people` class attribute for instances with the same `last_name`
   - return a collection of matching instances

Self Grading/Feedback

Unit tests have been provided in the bootstrap files that can be used to evaluate your solution. They must be run from the same directory as your solution.

```
$ rspec

John Smith
Jane Smith
....

Finished in 0.00436 seconds (files took 0.10999 seconds to load)
4 examples, 0 failures
```

A successful solution should have the following output.

```
$ ruby module2_lesson3_formative.rb

John Smith
Jane Smith
```

Submission

There is no submission required for this assignment but the skills learned will be part of a follow-on assignment so please complete this to the requirements of the unit test.