









```
employee.rb •
1 class Employee
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def pay
22    "Wooohooooo!"
23  end
24 end
25
```

```
hyena.rb •
1 class Hyena
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def feed(food)
22    if food == "crunchy bones"
23      "yipyipyip"
24    else
25      "grrrrrrrrrr"
26  end
```

```
vehicle.rb •
1 class Vehicle
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def fuel
22    "Vroooooooooom!"
23  end
24 end
```

```
1 module Memorable
2   module ClassMethods
3     def self.extended(base)
4       base.class_eval do
5         @all = []
6       end
7     end
8
9     def all
10      @all
11    end
12
13    def reset_all
14      self.all.clear
15    end
16
17    def count
18      @all.count
19    end
20  end
21
22  module InstanceMethods
23    def initialize
24      self.class.all << self
25      super
26    end
27  end
28 end
```

**Type:
Module**

**Code Name:
Memorable**

**Secret Power:
Saves the World from
Repetitive Code**

employee.rb

```
1 class Employee
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def pay
22    "Wooohooooo!"
23  end
24 end
25
```

hyena.rb

```
1 class Hyena
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def feed(food)
22    if food == "crunchy bones"
23      "yipyipyip"
24    else
25      "grrrrrrrrr"
26    end
27  end
28
```

vehicle.rb

```
1 class Vehicle
2   attr_accessor :all
3   @all = []
4
5   def initialize
6     self.class.all << self
7   end
8
9   def self.all
10    @all
11  end
12
13  def self.reset_all
14    self.all.clear
15  end
16
17  def self.count
18    @all.count
19  end
20
21  def fuel
22    "Vroooooooooom!"
23  end
24 end
25
```

```
employee.rb * hyena.rb * vehicle.rb *
1 require_relative '../models/cor...
2
3 class Employee
4   extend Memorable::ClassMethod
5   prepend Memorable::InstanceM...
6
7   def pay
8     "Wooohooooo!"
9   end
10 end
11
12
13
14
1
2
3 class Hyena
4   extend Memorable::ClassMethod
5   prepend Memorable::InstanceM...
6
7   def feed(food)
8     if food == "crunchy bones"
9       "yipyipyip"
10      else
11        "grrrrrrrrr"
12      end
13    end
14  end
1
2
3 class Vehicle
4   extend Memorable::ClassMethod
5   prepend Memorable::InstanceM...
6
7   def fuel
8     "Vroooooooooom!"
9   end
10 end
```



The space is
luxurious!

All Files (100.0%)

Generated 5 minutes ago

All Files (100.0% covered at 1.99 hits/line)

7 files in total. 99 relevant lines. 99 lines covered and 0 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
config/environment.rb	100.0 %	5	4	4	0	1.0
models/employee.rb	100.0 %	27	13	13	0	2.6
models/hyena.rb	100.0 %	26	13	13	0	3.0
models/vehicle.rb	100.0 %	26	13	13	0	2.6
spec/fixtures/employee_spec.rb	100.0 %	35	18	18	0	1.5
spec/fixtures/hyena_spec.rb	100.0 %	38	20	20	0	1.6
spec/fixtures/vehicle_spec.rb	100.0 %	34	18	18	0	1.5

Showing 1 to 7 of 7 entries

Generated by [simplecov v0.8.2](#) and [simplecov-html v0.8.0](#)
using RSpec

```
employee_spec.rb ×
```

```
1 require_relative '../spec/_  
2  
3 describe Employee do  
4   before(:each) do  
5     Employee.reset_all  
6   end  
7  
8   let!(:employee){Employee.  
9  
10  describe "Class methods"  
11    it "keeps track of the  
12      expect(Employee.all)  
13    end  
14  
15    it "can count how many  
16      expect(Employee.count)  
17    end  
18  
19    it "can reset the empl  
20      Employee.reset_all  
21      expect(Employee.count)  
22    end  
23  end  
24  
25  describe "Instance method  
26    it "can be initialized"  
27      expect(employee).to  
28    end
```

```
hyena_spec.rb ×
```

```
1 require_relative '../spec/_  
2  
3 describe Hyena do  
4   before(:each) do  
5     Hyena.reset_all  
6   end  
7  
8   let!(:hyena){Hyena.new}  
9  
10  describe "Class methods"  
11    it "keeps track of the  
12      expect(Hyena.all).to  
13    end  
14  
15    it "can count how many  
16      expect(Hyena.count).  
17    end  
18  
19    it "can reset the empl  
20      Hyena.reset_all  
21      expect(Hyena.count).  
22    end  
23  end  
24  
25  describe "Instance metho  
26    it "can be initialized"  
27      expect(hyena).to be_  
28    end
```

```
vehicle_spec.rb ×
```

```
1 require_relative '../spec_h  
2  
3 describe Vehicle do  
4   before(:each) do  
5     Vehicle.reset_all  
6   end  
7  
8   let!(:vehicle){Vehicle.  
9  
10  describe "Class methods" d  
11    it "keeps track of the  
12      expect(Vehicle.all).t  
13    end  
14  
15    it "can count how many  
16      expect(Vehicle.count)  
17    end  
18  
19    it "can reset the emplo  
20      Vehicle.reset_all  
21      expect(Vehicle.count)  
22    end  
23  end  
24  
25  describe "Instance method  
26    it "can be initialized"  
27      expect(vehicle).to be  
28    end
```

Keeping RSPEC Tests DRY

The DRYist Talk You Will Hear Today

By: David Coronado and Jessica Rudder

Create a New File

shared_examples_for_memorable.rb

Tells RSPEC this is
a file that will
contain shared
examples

Matches variable
used in spec to let
RSPEC know when
to use this file

Add Line to Class Spec File

it_behaves_like Memorable



Tells RSPEC to look
for a shared
examples file

Indicates it should
be the file ending in
_memorable

Move Memorable Tests to New File

```
describe Hyena do
  before(:each) do
    Hyena.reset_all
  end

  let!(:hyena){Hyena.new}

  describe "Class methods" do
    it "keeps track of the hyenas that have been created" do
      expect(Hyena.all).to include(hyena)
    end

    it "can count how many hyenas have been created" do
      expect(Hyena.count).to eq(1)
    end

    it "can reset the hyenas that have been created" do
      Hyena.reset_all
      expect(Hyena.count).to eq(0)
    end
  end

  describe "Instance methods" do
    it "can be initialized" do
      expect(hyena).to be_an_instance_of(Hyena)
    end
  end
end
```

Update to Implicit Subject

```
describe Hyena do
  before(:each) do
    described_class.reset_all
  end

  let!(:hyena) {described_class.new}

  describe "Class methods" do
    it "keeps track of the hyenas that have been created" do
      expect(described_class.all).to include(hyena)
    end

    it "can count how many hyenas have been created" do
      expect(described_class.count).to eq(1)
    end

    it "can reset the hyenas that have been created" do
      described_class.reset_all
      expect(described_class.count).to eq(0)
    end
  end

  describe "Instance methods" do
    it "can be initialized" do
      expect(hyena).to be_an_instance_of(described_class)
    end
  end
end
```



Change to an implicit subject using “described_class”

Update Variable Names

```
describe Hyena do
  before(:each) do
    described_class.reset_all
  end

  let!(:memorable) {described_class.new} ← Change from
  describe "Class methods" do
    it "keeps track of the hyenas that have been created" do
      expect(described_class.all).to include(memorable)
    end

    it "can count how many hyenas have been created" do
      expect(described_class.count).to eq(1)
    end

    it "can reset the hyenas that have been created" do
      described_class.reset_all
      expect(described_class.count).to eq(0)
    end
  end

  describe "Instance methods" do
    it "can be initialized" do
      expect(memorable).to be_an_instance_of(described_class)
    end
  end
end
```

Change from
specific “:hyena” to
“:memorable”

Update Your Test Labels

```
shared_examples_for Memorable do
  before(:each) do
    described_class.reset_all
  end

  let!(:memorable) {described_class.new}

  describe "Memorable class methods" do
    it "keeps track of the instances that have been created" do
      expect(described_class.all).to include(memorable)
    end

    it "can count how many instances have been created" do
      expect(described_class.count).to eq(1)
    end

    it "can reset the instances that have been created" do
      described_class.reset_all
      expect(described_class.count).to eq(0)
    end
  end

  describe "Memorable instance methods" do
    it "can be initialized" do
      expect(memorable).to be_an_instance_of(described_class)
    end
  end
end
```

Make it clear that these are tests for Memorable and be sure to remove any class specific references from the test labels

Type:

Shared Examples

```
require_relative './spec_helper'

shared_examples_for Memorable do
  before(:each) do
    described_class.reset_all
  end

  let!(:memorable) {described_class.new}

  describe "Memorable class methods" do
    it "keeps track of the instances that have been created" do
      expect(described_class.all).to include(memorable)
    end

    it "can count how many instances have been created" do
      expect(described_class.count).to eq(1)
    end

    it "can reset the instances that have been created" do
      described_class.reset_all
      expect(described_class.count).to eq(0)
    end
  end

  describe "Memorable instance methods" do
    it "can be initialized" do
      expect(memorable).to be_an_instance_of(described_class)
    end
  end
end
```

Code Name:

shared_examples_for_memorable

Secret Power:
Multiplies the
Power of 1 Test
Across Infinite
Classes

```
employee_spec.rb ×
```

```
1 require_relative '../spec/_  
2  
3 describe Employee do  
4   before(:each) do  
5     Employee.reset_all  
6   end  
7  
8   let!(:employee){Employee.  
9  
10  describe "Class methods"  
11    it "keeps track of the  
12      expect(Employee.all)  
13    end  
14  
15    it "can count how many  
16      expect(Employee.coun  
17    end  
18  
19    it "can reset the empl  
20      Employee.reset_all  
21      expect(Employee.coun  
22    end  
23  end  
24  
25  describe "Instance method  
26    it "can be initialized  
27      expect(employee).to  
28    end
```

```
hyena_spec.rb ×
```

```
1 require_relative '../spec/_  
2  
3 describe Hyena do  
4   before(:each) do  
5     Hyena.reset_all  
6   end  
7  
8   let!(:hyena){Hyena.new}  
9  
10  describe "Class methods"  
11    it "keeps track of the  
12      expect(Hyena.all).to  
13    end  
14  
15    it "can count how many  
16      expect(Hyena.count).  
17    end  
18  
19    it "can reset the empl  
20      Hyena.reset_all  
21      expect(Hyena.count).  
22    end  
23  end  
24  
25  describe "Instance metho  
26    it "can be initialized  
27      expect(hyena).to be_  
28    end
```

```
vehicle_spec.rb ×
```

```
1 require_relative '../spec_h  
2  
3 describe Vehicle do  
4   before(:each) do  
5     Vehicle.reset_all  
6   end  
7  
8   let!(:vehicle){Vehicle.  
9  
10  describe "Class methods" d  
11    it "keeps track of the  
12      expect(Vehicle.all).t  
13    end  
14  
15    it "can count how many  
16      expect(Vehicle.count)  
17    end  
18  
19    it "can reset the emplo  
20      Vehicle.reset_all  
21      expect(Vehicle.count)  
22    end  
23  end  
24  
25  describe "Instance method  
26    it "can be initialized"  
27      expect(vehicle).to be  
28    end
```

```
employee_spec.rb * hyena_spec.rb * vehicle_spec.rb *
1 require_relative '../fixtures'
2
3 describe Employee do
4   it_behaves_like Memorable
5
6 before(:each) do
7   Employee.reset_all
8 end
9
10 let!(:employee){Employee.new}
11
12 describe "Instance methods"
13   it "can be payed" do
14     expect(employee.pay).to
15     end
16   end
17
18 end
19
```

```
1 require_relative '../fixtures'
2
3 describe Hyena do
4   it_behaves_like Memorable
5
6 before(:each) do
7   Hyena.reset_all
8 end
9
10 let!(:hyena){Hyena.new}
11
12 describe "Hyena instance met
13   it "can be fed foods it do
14     expect(hyena.feed("apple"))
15   end
16   it "can be fed foods it lo
17     expect(hyena.feed("crunc
18   end
19 end
20
```

```
1 require_relative '../fixtures/s
2
3 describe Vehicle do
4   it_behaves_like Memorable
5
6 before(:each) do
7   Vehicle.reset_all
8 end
9
10 let!(:vehicle){Vehicle.new}
11
12 describe "Instance methods"
13   it "can be fueled up" do
14     expect(vehicle.fuel).to e
15   end
16 end
17
```



Now there's
room for two!



Employee

behaves like **Memorable**

Memorable class methods

 keeps track of the instances that have been created

 can count how many instances have been created

 can reset the instances that have been created

Memorable instance methods

 can be initialized

Instance methods

 can be payed

Hyena

behaves like **Memorable**

Memorable class methods

 keeps track of the instances that have been created

 can count how many instances have been created

 can reset the instances that have been created

Memorable instance methods

 can be initialized

Hyena instance methods

 can be fed foods it doesn't like

 can be fed foods it loves

Vehicle

behaves like **Memorable**

Memorable class methods

 keeps track of the instances that have been created

 can count how many instances have been created

 can reset the instances that have been created

Memorable instance methods

 can be initialized

Instance methods

 can be fueled up

All Files (100.0% covered at 3.92 hits/line)

8 files in total. 77 relevant lines. 77 lines covered and 0 lines missed

Search:

File	% covered	Lines	Relevant Lines	Lines covered	Lines missed	Avg. Hits / Line
config/environment.rb	100.0 %	8	6	6	0	1.0
models/concerns/memorable.rb	100.0 %	28	15	15	0	11.3
models/employee.rb	100.0 %	10	6	6	0	1.0
models/hyena.rb	100.0 %	14	8	8	0	1.1
models/vehicle.rb	100.0 %	10	6	6	0	1.0
spec/fixtures/hyena_spec.rb	100.0 %	20	11	11	0	2.0
spec/fixtures/support/shared_examples_for_memorable.rb	100.0 %	30	16	16	0	4.1
spec/fixtures/vehicle_spec.rb	100.0 %	17	9	9	0	2.0

Showing 1 to 8 of 8 entries





@davidjcoronado
@jessrudder



david.coronado@flatironschool.com
jessica.rudder@flatironschool.com