Lecture and yesterdays hmwk

Questions

Question Set 1

1. What is a class method?

**A class method is a method that operates on class objects rather than instances of the class**

1. How is a class method different from an instance method?

c. What's the difference between how you declare a class method versus an instance method?

Class methods are methods that are called on a class and instance methods are methods that are called on an instance of a class. Here is a quick example

class Foo

def self.bar

puts 'class method'

end

def baz

puts 'instance method'

end end

Foo.bar # => "class method"

Foo.baz # => NoMethodError: undefined method ‘baz’ for Foo:Class

Foo.new.baz # => instance method

Foo.new.bar # => NoMethodError: undefined method ‘bar’ for #<Foo:0x1e820>

1. What's the difference between how you call a class method versus an instance method?

**class method – Foo.bar ---- instance method – Foo.new.bar**

The benefit of this is 🡪

Question Set 2

Given the following code:

class User

attr\_accessor :name, :email

@@users = []

def initialize(name, email)

@name = name

@email = email

end

def self.all

@@users

end

def self.find(email)

@@users.find { |u| u.email == email }

end

def self.create(name, email)

u = User.new(name, email)

@@users.push(u)

u

end

end

Tip: If you'd like to quickly experiment with the code, try pasting the entire class definition above into a file named user.rb. You can then load up Pry or IRB and require\_relative 'user' to load your class in. Then, you can run the lines of code in questions 'd' and 'e' to use the class.

1. What functionality is the above code providing?

**adding users to a users array that is globally available**

1. What types of methods are find and create?

**class methods – because we don’t need instances to create**

1. How would you call the find and create methods?

**User.find --- Users.create**

d. What would you expect to happen if you instantiate a user and then call all on it? Like so:

u = User.new('Sally Forth', 'sally@example.com')

u.all

e. What happens when this code is run?

u = User.new('Sally Forth', 'sally@example.com')

u.class.all

Question Set 3

a. What are the names of two or more methods that are available on the Object class?

b. Since classes inherit from Object implicitly, what does it mean for all objects in Ruby that these methods are defined here?

Question Set 4

a. What happens when a subclass defines a method with the same name as its superclass?

b. How can the subclass of that superclass call upon the superclass' method?

**Topics:**

* OOP Review
* Stubbing methods in Rspec
* Intro to Modules and Mixins

Class is a blue print of a watch, but if 2 people have the same exact watch, these 2 are instances of the watch because they can both keep different values (when both are travelling) etc.

Houses – have 10 identical houses that aremade from the same blueprint (class) – but each house is different 🡪 they are instances.

A child inherits the class attributes form his parents, but as they move on in life they change. This is the same way where class A < B class A takes some or all of their attributes and methods.

there is only 1 class variable. 🡪 @@cars 🡪

Modules and Mixins

Rspec breakout.

first go to the directory – then -- bundle install

then bundle exec rspec spec/ (spec file #1).rb

\*\*if you see in the test that the before :each do – robot = Robot.new() has no parameters – we know that we don’t need parameters on the initisalize