Please Print Your Name: ANSWER KEY

Please Print Your CNetID:

---

1. (1 point) Describe one difference HTTP and HTML.

HTTP is a network protocol and describes a standard for accessing remote resources.
HTML is a markup language for use web browsers.

2. (2 points)
   A] Describe the difference between the `def` keyword and the `do` keyword.
   B] Provide a short example of how to use the `def` and `do` keywords.

   def is for defining methods; do is for defining an anonymous block of code.

```
def say_hello                        [1,2,3].each do |n|
  puts "Hello!"                        puts n
end                                  end
```

3. (1 point) Describe the main difference between a Ruby Array and a Ruby Hash.

   An array requires a zero-based integer index to retrieve a value
   A hash can use any kind of object as a key to retrieve a value

4. (2 points) Describe the difference between class methods and instance methods of a class. For example, how would you compare the "number_of_wheels" and "repair" methods in the class below?

```
class Bicycle
    def self.number_of_wheels

      # ...
    end

    def repair

      # ...
    end

end
```

Class methods are invoked against a class itself, and cannot access instance-level data.

Examples: Bicycle.number_of_wheels, Product.count

Instance methods are invoked against a specific instance, and can get/set instance-level data.

Examples: Bicycle.new.report, Product.new.title = "iMac"

5. **HTTP can be described as a....**

   a) a protocol used only by web browsers and servers

   b) a request protocol only

   (c) a request-response protocol

   d) a response protocol only

6. **HTTP requests can contain:**

   (a) a "headers" list of key-value pairs

   (b) a body that can contain text data

   (c) a resource locator

   d) a "footer" list of key-value pairs

7. **A web-based API....**

   a) always returns JSON, XML, or HTML

   (b) is a way for two programs to talk to each other

   c) can never be used by a web browser directly

   d) bypasses the typical HTTP requirements

8. **Browsers can send HTTP requests....**

   (a) using the GET method

   (b) using the POST method

   c) using the PUT method

   d) using the DELETE method

   e) browsers don't actually use any HTTP methods

9. **The command "rails server" will...**

   (a) start a local web server on port 3000

   b) deploy a web server to the internet

   (c) display a log of http traffic

   d) create a new rails application

10. **A Rails "route"...**

    a) connects controllers and views together

    b) can include querystring parameters

    (c) connects a url path to a Ruby method in your app

    d) must use lowercase characters only

11. **A Rails controller....**

    a) can contain no more than seven methods

    (b) derives ultimately from ActionController::Base

    (c) is responsible for handling HTTP requests

    d) must render a view template

12. **A Rails action...**

    (a) is a Ruby method

    b) can reside inside any class in your app

    c) must have a corresponding view template

    (d) is responsible for generating an HTTP response

13. **A Rails action...**

    a) can access querystring parameters with the `qrystring[]` hash

    (b) can inspect dynamic route segments

    (c) can receive form input field values

    d) should modify the `params[]` hash

14. **The following are valid inputs to the `render` method in a controller action:**

    a) render "<h1>Greetings</h1>"

    (b) render 'greetings'

    (c) render :json => ["hello", "goodbye"]

    (d) render

15. **A view template....**

    a) can access local variables defined in the corresponding action method

    (b) can access instance variables defined in the corresponding action method

    c) can access instance variables defined in any method of the same controller

    (d) can define its own local variables

    (e) can definite its own instance variables

16. A route can be named...

- (a) by using the **:as** option in the route definition line
- (b) to help DRY up controller and view code
- c) to generate three helper methods
- (d) differently from the name of the action

17. Rails "assets" are....

- (a) stylesheets
- (b) javascript files
- (c) images
- d) Ruby code that will never change
- e) static HTML files

18. form_for is different than form_tag because...

- (a) it will figure out the form **method** attribute value automatically
- (b) it will figure out the form **action** attribute value automatically
- c) it can be used for any kind of form you need
- (d) it should only be used for model-based forms

19. Models...

- (a) represent real-world things inside our software
- b) can access the **params[]** hash
- (c) can contain methods that you manually write
- d) always derive from ActiveRecord::Base
- e) can be associated to at most one other model
- f) can render views
- (g) should use the singular noun form, i.e. **Car**, instead of the plural noun form, i.e. **Cars**
- h) can not be accessed from views
- i) can contain action methods if the route specifies the model class using the **:controller =>** option
- (j) are often database-backed
- k) allow mass-assignment to all columns by default

20. RESTful routing...

- a) expects the intended action to be in the URL path string
- (b) leverages the HTTP notion of resources
- c) always requires seven actions
- d) provides automatic security features
- (e) defines conventions for all dynamic segments
- f) requires that your route names must match your model names

21. The following are Rails features that intentionally embrace the DRY principle

- (a) partials
- b) models
- c) rake routes
- (d) layouts
- e) embedded ruby syntax

22. Data validations are best implemented by...

- a) the controller
- (b) the model
- c) the view
- d) the command line
- e) rails console

23. The one primary responsibility of a view is to...

- a) DRY up your code
- b) generate JSON
- (c) apply markup to controller-provided data
- d) retrieve data from the database

24. Associations are....

- (a) implemented in models
- b) implemented in controllers
- (c) used by models
- (d) used by controllers
- e) used in routes

25. **One-to-Many associations require...**

    a) that each model contain a primary key

    b) that only one model contain a foreign key

    c) a belongs_to declaration

    d) a has_many declaration

26. **belongs_to ...**

    a) generates a foreign key column

    b) generates an instance method

    c) is contained in the model without the foreign key

    d) is contained in the model with the foreign key

27. **has_many**

    a) generates a foreign key column

    b) generates an instance method

    c) is contained in the class without the foreign key

    d) is contained in the class with the foreign key

28. **Many-to-many associations....**

    a) always require only two tables

    b) always require at least three tables

    c) requires at least two foreign keys in one model

    d) cannot be achieved in Rails

29. **Validation rules are triggered by calls to ....**

    a) .valid?

    b) .where

    c) .save

    d) .find_by_id

    e) .new

    f) .errors

    g) .create

    h) .update_attributes

30. **A Rails migration is best defined as...**

    a) the schema of the entire database

    b) the schema for one table

    c) SQL instructions for at least one database schema change

    d) a Ruby script that executes instructions for at least one database schema change

31. **In order to define a home page in a Rails application that uses the MVC pipeline, ...**

    a) the default public/index.html file must be deleted

    b) the **root** method must be called inside routes.rb

    c) a route equivalent to **GET "/"** must be defined

    d) you must define a HomeController class

32. **Routes defined in routes.rb ....**

    a) are evaluated using a best-match algorithm upon receiving an http request

    b) are evaluated top-down for the first possible match upon receiving an http request

    c) are uniquely defined by the URL path

    d) are uniquely defined by the combination of the http method and the URL path

    e) can contain multiple dynamic segments

    f) must use **:id** as a dynamic segment

33. **The best way to determine the actual set of routes defined by a Rails application is...**

    a) **rake routes**

    b) viewing the routes.rb file

    c) watching the server log

    d) **rake db:migrate**

    e) **rake about**

34. Controller instances....

a) are held in memory for the life of the application

b) are held in memory until the action method returns

(c)) are held in memory until an HTTP response has been generated

(d)) are created by Rails

e) are created by the routes file

f) are created by code you must write yourself

35. The following command:

rails generate model Books title:string isbn:string

(a)) generates a model class file

(b)) generates a migration file

c) executes the migration file

(d)) makes the title and isbn attributes available for mass assignment

e) properly adheres to Rails naming conventions

36. rails console....

(a)) is similar, but different from, irb

b) is exactly the same as irb

c) shows the server log

d) is a simulator for making http requests

37. In the following standalone Ruby 1.9 script:

```
require 'json'
require 'open-uri'
url = "http://mydomain.com/hello.json"
x = JSON.parse(open(url).read)
```

a) x is an instance of class Array

b) x is an instance of class Hash

c) x is an instance of class String

(d)) it is not possible to infer the type of x from this code example

(e)) url is an instance of class String

(f)) both **require** statements are necessary

g) only **require 'json'** is necessary

h) only **require 'open-uri'** is necessary

i) none of the **require** statements are necessary

38. The HTML generated by both form_tag and form_for is preferred over manually writing a <form> tag because...

(a)) they adhere to **HTML5** standards for element syntax

(b)) they generate an authenticity token inside of a hidden field to provide CSRF protection

(c)) they can generate a hidden field to simulate an HTTP method other than GET or POST

d) they apply built-in CSS styles automatically

(e)) they can make your source code easier to read

39. In order to work, a form requires a minimum of...

a) two actions *(circled)*

b) one partial

c) one HTML <form> element *(circled)*

d) two controllers

e) one action

f) two routes *(circled)*

g) one route

40. This code...

```
resources :products
```

a) generates 4 total routes

a) generates 7 total routes *(circled)*

b) generates 4 named routes *(circled)*

c) assumes a controller named "ProductsController" *(circled)*

d) assumes a controller named "ProductController"

41. "Mass assignment" is triggered by the following ActiveRecord::Base methods:

a) create *(circled)*

a) new *(circled)*

b) destroy

c) update_attributes *(circled)*

d) save

42. Given these modeling requirements:

1. The CTA transit system has seven train lines
2. A train line has many stations
3. A train line stops at every station on its line
4. A train line travels its full length 14 times in a 24-hour period

Provide a class model diagram in the space below, showing the names of the classes and cardinality among them (1-to-1, 1-to-many, or many-to-many):

TrainLine —1-to-Many→ Stop

TrainLine ←Many-to-Many→ Station

Station —1-to-Many→ Stop
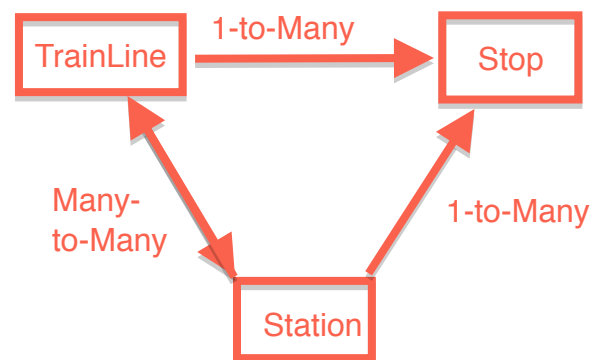
DEBUGGING EXERCISES

You've been handed some code to debug. For each action below, list each error that the user would encounter and what code you would change, and in which file(s), in order to resolve it. There are at least 4 bugs to resolve in each of the two actions.

1. (10 points) A user goes to http://ourdomain.com/users/4 in their browser. Assume that a user with the ID number 4 exists. We want to show the user's profile and a list of pictures belonging to that user.

   Missing action 'details'.
   Change 'Users#details' to 'Users#show' in routes.rb. (2 points)

   Undefined method 'pictures' for <#User> in users_controller.rb.
   Add has_many :pictures to user.rb. (4 points)

   Undefined local variable 'pictures' in show.html.erb.
   Change to @pictures. (2 points)

   Undefined method 'url' for <#Picture> in show.html.erb.
   Change to .source. (2 points)

2. (15 points) A user goes to http://ourdomain.com/pictures/1/edit in their browser. Assume that a picture with the ID number 1 exists. We want to show a form to edit the picture. Once they click submit, the picture should be updated and the user should be redirected to the profile of the user that picture is associated with.

    No route for GET '/pictures/1/edit'.
    Change put '/pictures/:id/edit' to get '/pictures/:id/edit' in routes.rb. (2 points)

    No route for PUT '/pictures/1/edit'.
    Change post '/pictures/:id' to put '/pictures/:id' in routes.rb. (2 points)

    No error messages, but picture columns will all be set to nil since top level keys don't exist in the params hash when using form_for.
    In pictures_controller.rb, either change lines 11-13 to params[:picture][:whatever] or change line 14 to @picture.update_attributes(params[:picture]) (7 points)

    Undefined method 'user' for <#Picture> in pictures_controller.rb.
    Add belongs_to :user to picture.rb. (4 points)