**PART I: ARCHITECTURE**

1. What is the difference between an HTTP PUT request and a POST request?
   * HTTP PUT puts a resource at a very specific URI, and will replace it if there is already one at the URI. It will create a new one if there’s not one there already.
   * POST sends resources to the URI and expects the resource at the URL to figure out how to handle it.
   * The difference between the two is that POST identifies a resource that will handle the enclosed identity, but can be applied in several ways. However, a PUT request must not attempt to apply the request to another resource other than the one intended.

<a href="images/new?request\_type=PUT" method="POST">create a new image</a>

1. Is the target URL relative or absolute?
   * Relative.
2. What is the difference between an absolute and relative URL?
   * If it were absolute, the entire url (<http://www.example.com/images/new?request_type=PUT>) would be used. This is used because it is possible to use between several different projects without having to change the markup.
3. If you clicked on this link, what kind of request would your browser generate? (which HTTP method?) Assume no JavaScript modifies the behavior of the link.
   * A GET request.
4. Is there a querystring? What is it?
   * A QueryString collection isused to retrieve the variable values in the HTTP query string. It is specified by values following the “?” character. There is one present in this URL, and it contains “request\_type=PUT”.
5. What is lacking from the link declaration that would otherwise enhance accessibility?
   * Adding :title to the url would increase accessibility.
6. What are the roles of the database and Web browser in most Web applications? (One sentence for each.)
   * The database in a web application serves to sort, filter, and present data based on requests from users.
   * The web browser in a web application serves to display the HTML code sent from the web server.

Given the following HTTP response header:

HTTP/1.1 200 OK  
Date: Wed, 09 Mar 2011 16:43:33 GMT  
Server: Apache  
Connection: Keep-Alive  
Keep-Alive: timeout=2, max=100  
Etag: "110e412f-7df-49e0f6a106500"  
Vary: Accept-Encoding

1. Would an HTTP response that begins like that usually contain a body? Why or why not?
   * Yes, unless it is a HEAD request in which case the requested resources are sent in the response without any message-body.

**PART II: RUBY**

1. Write a Ruby class definition that meets the following criteria:
   * class is called Troll
   * class has publicly accessible attributes ugliness, smelliness, and strength
   * upon instantiation, an object of this class has a member variable, a String, called grunt, whose initial value is "UNGAH" (that's pronounce "oon-guh").
   * class has an instance method called speak() that prints the value of the instance variable grunt 42 times
   * class has an instance method called reverse() that prints the value of the instance variable grunt backwards
   * class has a static/class method called propogate(), which returns a Troll instance whose grunt attribute is "eegah"

class Troll

attr\_reader :ugliness, :smelliness, :strength

attr\_writer :ugliness, :smelliness, :strength

attr\_accessor :ugliness, :smelliness, :strength

def initialize(grunt = “UNGAH”

@grunt = grunt

@ugliness, @smelliness, @strength

end

def speak

(1..42).each { |i| puts @grunt }

end

def reverse

puts grunt.reverse

end

def self.propogate

Troll.new("eegah")

end

end

1. Imagine a Troll instance fred, which, when the following method is called:

fred.respond\_to?("fight")

returns true. What is missing from your class definition in order for this example to be accurate?

* + def respond\_to?(:action)

if action == "fight" then

return true

end

return false

end

1. Does the respond\_to?() method illustrate object-oriented polymorphism? If so, in what manner?
   * Polymorphism in object oriented languages is the ability to create a variable, function or an object with more than one form. This means that using one type, another can be derived. This is more simply defined as that, from one message, many different actions can be taken.

The answer to the question is yes. This is because the message “fight” in this case is interpreted by the Troll class as true. However, the “fight” message might be interpreted by objects of the Elf class as false, resulting in a much different “reaction” of the objects of that class.

1. According to Ruby conventions, what kind of value would you expect to receive from a method that ends in a question mark (?) ?
   * The question mark character has no meaning in the Ruby interpreter. However, according to the naming conventions of the language, a method whos name ends with a ? returns a value that answers the question with a Boolean. This means that you should expect a true or false “answer” to the question.
2. According to Ruby conventions, what is the difference between pairs of methods like do\_this and do\_this! (notice the bang) ?
   * I did notice the bang. After running, the methods without the bang will return a new object, reflecting the result of the action. The bang version of the same method does everything the non-banged method does except its result isn’t a new object, it instead performs those actions on the original object. This is considered dangerous because it replaces the original with the new version.
3. Briefly explain Ruby's type system. What is it (by name)? What does it mean?
   * Quack Quack. Ruby employs what is called a “Dynamic Typing System” or “Duck Typing” to the more hilarious programmers. Similar to that used in JavaScript, Lua, MATLAB and Perl, this system performs most of its type-checking at run-time instead of at compile-time (such as Java, C++, and PASCAL). Values have types, but variables do not. This means that variables can refer to a value of any type.
4. What type of Ruby object does the following expression yield?

%w( master rails and then try another framework you'll never go back)

* + An array of string elements

1. Given an array of strings called @happy\_places, would these two snippets of code do the same thing?

@happy\_places.each do |happy\_place|  
  puts happy\_place  
end

and

@happy\_places.each {|hp| puts hp}

* + Yes, they are the exact same thing.

1. Given a function that needs to return a value to its caller, does the function need an explicit return statement? If so, explain why. If not, then what can you always expect a Ruby function to return?
   * In short, no. If there is no explicit return statement in a Ruby method, then the value of the last statement. However, you can have an explicit return statement to make sure that what you want returned returns, instead of simply returning the last statement calculated.

**PART III: RAILS**

1. Name four ActiveRecord callbacks that you can bind methods to.
   * Before\_validation
   * Before\_save
   * Before\_create
   * Before\_destroy
2. The Rails convention maps HTTP methods to certain controller methods, and those methods usually involve specific CRUD operations on models. Given the following CRUD database methods:  
   create, read, update, and delete  
   and the following HTTP methods:  
   GET, PUT, POST, DELETE  
   and the following controller actions:  
   index, new, create, edit, update, destroy Complete the following table.

|  |  |  |
| --- | --- | --- |
| HTTP Method | Controller Action | CRUD Operation |
| GET | Index | read |
| GET | New | read |
| POST | Create | create |
| GET | Edit | read |
| PUT | Update | update |
| DELETE | Delete | delete |

1. Rails "simulates" PUT and DELETE requests. Why?
   * This is because PUT and DELETE are usually supported by HTML. They are usually “simulated” by abusing PUT and DELETE.
2. What is the difference between the two Rails environments 'production' and 'development' ?
   * The major differences between production and development environments are performance and stability. While producing, models are cached in memory, so that once they have been read once, the files don’t have to be read again which assist in speeding up production. Conversely, a development environment will reload the models each time it receives a request. The main difference between the two is how the models are accessed while initializing.
3. Usually, Rails controllers incorporate plural nouns, such as ProtestsController and RevolutionsController. In what case should a controller have a singular name like GeocodingController?
   * Never. This is because it is the RESTful convention to pluralize. Additionally, rails automatically looks for the pluralized name for that controller when it goes to reference it so it will break your project.

However, if there is only going to be one record in the database you can singularize, it is just not recommended.

1. What is a Rails "helper method" and when should they be defined and used by you, the developer?
   * A Rails “helper method” is a small section of code that can be called into views, and allow the developer to not have to repeat himself (DRY). They should be used when you find you’re re-using code over and over again in your views. I found I used it when I was writing the code to allow formatting of the date in the Articl.es project (4 &5).

Assume you have a Flower AR class that has\_and\_belongs\_to\_many :bees, and a Bee class that has\_and\_belongs\_to\_many :flowers.

1. What must exist in the database schema in order for AR to infer the proper foreign key / relationship?
   * There must be a column in both the bee and flower DB such as flower\_id and bee\_id respectively. This way they can be linked.

Assume that a Bee :belongs\_to a Hive and a Hive has\_many :bees. Also assume a GET request is sent to the FlowersController#show action, which contains a finder method call @flower = Flower.find(params[:id]. Assume the view app/views/flowers/show.html.haml displays the name of the Flower and each Flower's bee's name and Hive name like so:

- @flower.bees.each do |b|  
  %h1= b.name  
  %p= b.hive.name

If you were tailing the log of your application during the rendering of the response, you would notice tons of database queries.

1. Are all of those queries ok? If so, explain why. If not, explain how you would reduce the number of database queries (without hand-rolling your own SQL query).
   * No, you can reduce the number of queries you have by using eager loading. I would :include => :flower, [hive] to the access, and therefore cut down on the SQL queries by a great deal.