**Architecture**

1. *What is the difference between HTTP PUT and HTTP POST requests?*
   1. **A PUT request is like a file upload. You are putting an item at the given URI, without expecting any kind of action from the server, aside from writing whatever information is contained in the body of the PUT request. If the resource being written to doesn’t exist, the PUT request creates it.**
   2. **A POST request initiates action on the server side. It could be a request to write to a specified URI, or it could be a request to write to a different URI, or to append a comment on a different page. It’s all up to how the request is formatted, and how the server has been programmed to handle that specific request.**
2. *Is the URL in the provided line of code relative, or absolute?*
   1. **The URL is relative.**
3. *What is the difference between an absolute and a relative URL?*
   1. **An absolute URL is one that points exactly to a source; I.E., http://buffalocomputerconsulting.com/images/computer.jpg is an absolute path to the image ‘computer.jpg.’ A relative URL is one that points to a resource relative to the place the current page is located; IE, ‘images/computer.jpg’ specifies that the file ‘computer.jpg’ is within a subfolder of the current directory, named ‘images/’.**
4. *If you clicked on this link, what kind of request would your browser generate?*
   1. **The browser would generate a POST request, as specified by the method attribute.**
5. *Is there a query string? If so, what is it?*
   1. **Yes, there is a query string; it is ‘?request\_type=PUT’.**
6. *What is lacking from the link declaration that would otherwise enhance accessibility?*
   1. **A title attribute on the link with useful information about the link is missing; it would enhance accessibility.**
7. *What are the roles of the database and Web browser in most Web applications? (One sentence for each.)*
   1. **The database is the centralized storage area for items used in the application; it stores information about each of your objects, such as an item’s price, image, and description in a shopping application.**
   2. **The Web browser renders the information that the application sends to it; all of the styling, positioning, and elements displayed on the page are put together by the browser, so that the user making use of the application can interact with it properly.**
8. *Would this type of response usually contain a body? Why or why not?*
   1. **I would say no; the response is missing information having anything to do with content. Also, the connection type specifies that the connection should stay open for further requests.**

**Ruby**

1. *Write a Ruby class definition that meets the criteria listed.*
   1. **class Troll  
       attr\_writer :grunt  
       attr\_reader :ugliness, :smelliness, :strength, :grunt  
       @ugliness  
       @smelliness  
       @strength   
       @grunt = "UNGAH"  
       def initialize(grunt)  
       @grunt = grunt  
       end  
       def speak  
       for each in 1..42 do   
       puts @grunt  
       end  
       end  
       def reverse  
       puts @grunt.reverse  
       end  
       def self.propogate  
       Troll.new.grunt = "eegah"  
       end  
      end**
2. *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?*
   1. **My class definition doesn’t have a method named respond\_to, which would match whether or not the given string (case-insensitive) matches a list of things (also strings) that Fred responds to.**
3. *Does the respond\_to?() method illustrate object-oriented polymorphism? If so, in what manner?*
   1. **Yes. You can check if Fred responds to the string, or you can “tell” him to, with respond\_to!.**
4. *According to Ruby conventions, what kind of value would you expect to receive from a method that ends In a question mark?*
   1. **I would expect to receive a Boolean response; true or false.**
5. *According to Ruby conventions, what is the difference between pairs of methods like do\_this and do\_this! (notice the bang?)*
   1. **Do\_this will evaluate as long as conditions are met. Do\_this! Evaluates regardless; kind of like executing a command as root, or passing a unix-like system a --force parameter – basically, “do this regardless of what the repercussions are” as opposed to “do this if everything is set up properly.”**
6. *Briefly explain Ruby’s type system. What is it (by name)? What does it mean?*
   1. **Ruby’s type system is called a dynamic type system. It means that type-checking is done at runtime, and that variables do not have types; values do.**
7. *What type of Ruby object does the following expression yield? %w( master rails and then try another framework you’ll never go back)*
   1. **%w() creates an array of words, separated by white space.**
8. *Given an array of strings called @happy\_places, would these two snippets of code do the same thing?*
   1. **Yes. These are different only in that one is encased in a do block, and one is encased in curly braces. The logic behind them should be the same.**
9. *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?*
   1. **Ruby does not require an explicit return statement. Ruby functions will always return the last evaluated statement.**

**Rails**

1. *Name four ActiveRecord callbacks that you can bind methods to.*
   1. **You can bind methods into before\_validation, before\_create, validate, and before\_save, to name four.**
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, 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** |
| **PUT** | New | **Create** |
| **PUT** | Create | **Create** |
| **POST** | Edit | **Update** |
| **PUT** | Update | **Update** |
| **DELETE** | Delete | **Delete** |

1. *Rails “simulates” PUT and DELETE requests. Why?*
   1. **Rails can simulate browser requests so that a Web developer can test the functionality of their site without having to test it through a browser.**
2. *What is the difference between the two Rails environments ‘production’ and ‘development’ ?*
   1. **The Rails development environment is configured such that development is made easier. Changes to the code are gotten each time a get request is sent to the server; you don’t have to restart your application every time you make a change to your code. It may run a bit slower this way, but a bit of a sluggish response time when in development goes a long way to prevent catastrophic crashes when the site is put into production.**
   2. **The production environment is all about speed; resource caching is enabled, in order to get the page to the user as fast as possible. In order for a code change to take effect, the entire application must be restarted. Ideally, you’ve worked out all the bugs while in the development environment, so that the production application doesn’t have to be restarted, but a short restart is always preferable to a cataclysmic crash.**
3. *Usually, Rails controllers incorporate plural nouns, such as ProtestsController and RevolutionsController. In what case should a controller have a singular name like GeocodingController?*
   1. **In the case where you have a singular resource, such that an ID does not have to be referenced, a singular name would be used. (IE, if a logged-in user was going to /profile on the website, the application would display the logged in user’s profile.)**
4. *What is a Rails “helper method” and when should they be defined and used by you, the developer?*
   1. **A Rails “helper method” is to Rails what a function is to other programming languages; a place to put code that would otherwise be repeated all over the place, so that code doesn’t have to be copied+pasted (DRY). As a developer, I should define and use them when I know there is going to be a common piece of code called throughout my application.**
5. *What must exist in the database schema in order for the AR to infer the proper foreign key / relationship?*
   1. **You would need an indication in the schema that another database must be referenced; either a new column in the existing database with the ID of the external party, or a join table if there are many possible external parties per entry in the first database.**
6. *Are all of the queries listed okay? If so, explain why. If not, explain how you would reduce the number of database queries (without hand-rolling your own SQL query).*
   1. **For a small enough database, these queries would be okay, although not nearly optimized. To make sure that expanding data doesn’t kill the response time of my application, I would reduce the number of queries by joining the tables by ID, and then running a single query to get both the bee and hive name associated with that ID.**