Joe Zeimen
Web Apps Midterm

Architecture

1. What is the difference between an HTTP PUT request and a POST request?

A put request was created so that the body of the request would be stored at that specific location in the URI, a post request expects the web server to figure out what to do with the data it just received.

You see a link in an interface whose markup is as follows:

```
<a href="images/new?request_type=PUT" method="POST">create
a new image</a>
```

2. Is the target URL relative or absolute?
   Relative

3. What is the difference between an absolute and relative URL?
   Absolute specifies the protocol as well as the hostname
protocol://www.hostname.com/path/to/resource

4. 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.

POST

5. Is there a querystring? What is it?
   Yes everything after the "?" and before the quotes is the query string.
"request_type=PUT" This tells data is to be passed to the web application located at the relative address.

6. What is lacking from the link declaration that would otherwise enhance accessibility?
   A title so that if someone hovers over it with a mouse, it will tell them what it will do.

7. What are the roles of the database and Web browser in most Web applications? (One sentence for each.)

The web browser displays the html/css/javascript that it queried from a server; it also maintains cookies.
A database stores the data a web app needs to work, persistence.

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
```

8. Would an HTTP response that begins like that usually contain a body? Why or why not?

Yes it has an Etag which is usually a hash of the document that it will send.

Ruby

9. 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_accessor :smeliness, :ugliness, :strength
      def initialize
         @grunt = "UNGAH"
      end
      def speak
         42.times {puts @grunt}
      end

      def reverse
         puts @grunt.reverse
      end

      def propagate
         t = Troll.new
         t.set_grunt  "eegah"
         t
      end
      def set_grunt string
         @grunt = string
      end
```

```
end
```

10. 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?

The class needs a method called `fight`

11. Does the `respond_to?()` method illustrate object-oriented polymorphism? If so, in what manner?

No, this is demonstrating reflection.

12. According to Ruby conventions, what kind of value would you expect to receive from a method that ends in a question mark (?) ?

A boolean

13. According to Ruby conventions, what is the difference between pairs of methods like `do_this` and `do_this!` (notice the bang) ?

The ! means that the object that you called it on will be updated. without the ! the method will return a new object with the operation. For example a .sort on an array will return a new sorted array. .sort! will sort the array in place.

14. Briefly explain Ruby's type system. What is it (by name)? What does it mean?

It is dynamic typing. It means you can switch they type of a variable whenever you want.

15. What type of Ruby object does the following expression yield? `%w( master rails and then try another framework you'll never go back)`

It creates an array where each word is its own element

16. 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 would both do the same thing

17. 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?

No, whatever object is the result of the last statement in the function is returned

Rails

18. Name four ActiveRecord callbacks that you can bind methods to.

before_validation
after_validation
before_save
around_save

19. 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	N/A
POST	create	Create
Get	edit	Read
Put	update	Update
Delete	delete	Delete

20. Rails "simulates" PUT and DELETE requests. Why?

Because, it is storing its own data and it doesn't have "files" that it can just update or delete so it accepts the request and processes it in a meaningful way.

21. What is the difference between the two Rails environments 'production' and 'development' ?

Each use their own database.
Development – checks for changes for every request
Production – uses caching to speed up requests

22. Usually, Rails controllers incorporate plural nouns, such as `ProtestsController` and `RevolutionsController`. In what case should a controller have a singular name like `GeocodingController`?

When there is a special object such as the logged in the user, this is something that is frequently looked up without a reference id.

23. What is a Rails "helper method" and when should they be defined and used by you, the developer?

A rails helper method is a function that performs very common tasks such as pluralization. You should make your own when rails doesn't have one and you have repeated code you could dry out.

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.

24. What must exist in the database schema in order for AR to infer the proper foreign key / relationship?

There must be a join table called bees_flowers with foreign keys on each row, one for a bee and one for a flower.

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.

25. 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).

Use eager loading. Add the argument “include: [:hive]” when searching for all the bees.