QUIZ 2: RAILS, WEB ARCHITECTURE AND BEHAVIOR DRIVEN DESIGN

This quiz covers content from the first four weeks of lectures, slides and readings, with an emphasis on the material from weeks 3 and 4. The quiz is designed to take less than an hour, and should be taken without the aid of a Ruby interpreter.

Please note that you are allowed only one attempt per question. You can review the "Check" and "Save" functionalities in the edX introduction (/courses/BerkeleyX/CS169.1x/2013_Spring/courseware/Overview/edx_introduction/3).

QUIZ 2-1

Which of the following are goals of Behavior-Driven Design (BDD)? Select all that apply.

- ☑ To verify that the application meets the specification
- ☑ To validate that the design does what the customer wants
- ☑ To help the customer understand the use of the application
- ☑ To ask questions about the behavior of an application before and during development

You have used 1 of 1 submissions

QUIZ 2-2

A ----- refers to a single ----- with one or more -----, each of which is composed of a sequence of -----.

- ○ feature, step, user stories, scenarios
- ○ feature, scenario, steps, user stories
- ○ user story, feature, scenarios, steps
- ○ user story, scenario, features, steps

[**Final Check**]   [**Save**]   *You have used 0 of 1 submissions*

QUIZ 2-3

The purpose of the Lo-Fi UI and storyboards is to debug the UI before you program it.

○ True

○ False

Final Check **Save** *You have used 0 of 1 submissions*

QUIZ 2-4

Which of the following sets of code will add a new movie entry to the database with all attributes properly set? Select all that apply.

☑
```
batman_begins = Movie.create(:title => 'Batman Begins', :release_date => '6/15/2005',
:rating => 'PG-13')
```

☑
```
dark_knight = Movie.new
dark_knight.title = 'The Dark Knight'
dark_knight.release_date = '7/18/2008'
dark_knight.rating = 'PG-13'
dark_knight.save!
```

☑
```
dark_knight_rises = Movie.create
dark_knight_rises.title = 'The Dark Knight Rises'
dark_knight_rises.release_date = '7/20/2012'
dark_knight_rises.rating = 'PG-13'
```

You have used 1 of 1 submissions

QUIZ 2-5

A user submits a new movie form. How would the associated controller access the title of the movie?

○ session['title']

○ session.title

◉ params['title']

○ params.title

○ All of the above

You have used 1 of 1 submissions

QUIZ 2-6

Which of the following is an invalid statement regarding `render` and `redirect_to` ?

○ `render` doesn't send a new request but `redirect_to` does

◉ `redirect_to` doesn't send a new request but `render` does

○ `render` will not cause the current action to stop executing

○ `redirect_to` will not cause the current action to stop executing

QUIZ 2-7

Which of the following can be used to store state ACROSS requests from the same user? Select all that apply.

- ☑ Cookies
- ☑ The `session[]` hash
- ☐ Instance variables set by a controller method
- ☑ Class variables set by a controller method

QUIZ 2-8

What does line 4 do in the following code? Select all that apply.

```
1 class PostsController < ActionController::Base
2 def add_post
3 # ...code to create new blog post...
4 flash[:notice] = "Post added successfully"
5 redirect_to posts_url
6 end
7 end
```

- ☐ Writes a message to the log
- ☑ Makes the message available to whatever view is rendered for this action
- ☑ Persists the message until the next request

QUIZ 2-9

If a user enters values in a view, and then those values get sent to the controller as `params['movie']['title']`, those values are accessible to the controller action during

- ⦿ this request only
- ◯ this request and the next request
- ◯ this request and all subsequent requests

QUIZ 2-10

If a controller action sets an instance variable, the instance variable's value is accessible to

- ⦿ views rendered for this request only
- ◯ views rendered for this request and the next request
- ◯ views rendered for this request and all subsequent requests

QUIZ 2-11

Sally wants her website to have a special layout on the first Tuesday of every month. She has the following controller and test code:

```
# HomeController
def index
  if Time.now.tuesday?
    render 'special_index'
  else
    render 'index'
  end
end

# HomeControllerSpec
it "should render special template on Tuesdays" do
  get 'index'
  if Time.now.tuesday?
    response.should render_template('special_index')
  else
    response.should render_template('index')
  end
end
```

What FIRST principle is not being followed?

- ○ Fast
- ◉ Independent
- ○ Repeatable
- ○ Self-checking
- ○ Timely

QUIZ 2-12

Suppose RottenPotatoes.com wants to launch m.rottenpotatoes.com, an alternate version of the site for mobile clients. If the site was built using an MVC architecture, the majority of the work will go into writing new

- ◉ views
- ○ models
- ○ controllers
- ○ schema

QUIZ 2-13

Why do software projects fail? Select all that apply.

- ☑ They don't do what customers want
- ☑ They are late

☑ They are over budget

☑ They are hard to maintain and evolve

You have used 1 of 1 submissions

QUIZ 2-14

Which of the following are true about user stories? Select all that apply.

☑ They should describe how the application is expected to be used

☑ They should have business value

☐ They do not need to be testable

☐ They should be implemented across multiple iterations of the Agile lifecycle

You have used 1 of 1 submissions

QUIZ 2-15

Which of the following is true about implicit requirements and explicit requirements?

○ You cannot write user stories for both explicit and implicit requirements

○ Implicit requirements tend to be more concise, while explicit requirements tend to be more verbose

○ Implicit requirements are the logical consequence of explicit requirements, and typically correspond to integration tests

◉ Explicit requirements are usually defined with imperative scenarios and implicit requirements are usually defined with declarative scenarios

You have used 1 of 1 submissions

QUIZ 2-16

When should you write specific (imperative) user stories instead of declarative user stories?

○ When you want to express the details of the user interface for business value or customer need

○ When you want to write user stories for both explicit and implicit requirements

○ When you want to write simpler, less verbose, and more maintainable user stories

○ Never -- you should always write declarative user stories

Final Check   **Save**   *You have used 0 of 1 submissions*

(http://www.facebook.com/EdxOnline) (https://twitter.com/edXOnline)