**IT 231 – Final Exam  
November 18, 2010**

**Part A. Computer Problems.** Create Rails items according to these specifications. Submit this project in a zip file on the COL website. Choose one of the options below. Indicate which option you choose and which enhancements you choose to implement.  
  
Option 1 \_\_\_\_\_ Option 2 \_\_\_\_\_ Specify titles and headings in controller \_\_\_\_\_

Validations \_\_\_\_\_ Input gender as radio buttons or dropdown menu \_\_\_\_\_

Enhancements to CSS style\_\_\_\_\_

**Option 1, 60 points:** Create a rails project named Company using the scaffold software. The website should manage resources named Employee, with fields name (string), gender (string), empid (string), salary (integer).

**Option 2, 80 points:** Create a rails project named Company **WITHOUT** using the scaffold software. Generate a controller named EmployeeData with two views named enter and display. Create a model named Employee with fields fields name (string), gender (string), empid (string), years with company (integer). Use FormTagHelper or FormHelper controls, with text to label the controls to make them easy to understand for the user. Use a layout file named EmployeeData for everything common to both views.

The enter view should contain text fields for entering the name, gender, id, and years and a submit button with the caption Enter EmployeeData. The display view should show all of the employee data currently in the database. Also include a layout file named employee\_id.html.erb that includes everything common to the two views.

**Specify titles and headings in controller, 5 points:** Create a variable @title in each controller action that specifies the title and h1 header on the layout.

**Validations, 5 points:** Put validation statements in the model file (employee.rb) that insure that name is nonempty, gender is either M or F, ID is of the form EAC-3829 (3 uppercase letters, a dash, and 4 digits). Years with the company should consist of 3 digits. Use **validates\_presence\_of** for name and **validates\_format\_of** with regular expressions for gender and empid. Don’t validate the gender if you use radio buttons or a dropdown menu. Note: without a scaffold, the error messages for validation will not appear without some extra work. The validation, however, does prevent illegal input from being saved in the database.

**Input gender with radio buttons or a dropdown menu, 5 points:** Either use radio buttons labeled as Female and Male, or use a dropdown menu with entries Female and Male.  
 **Modify or add a CSS stylesheet, 5 points**: Modify or add a stylesheet that does the following: sets the color to silver, the font to Georgia, and the background image to backimage.jpg. Use the hex color code for silver; use this property for the background image:   
**background-image: url(/images/backimage.jpg);**

**Part B: Multiple Choice Questions.** Choose the correct answer for each question. Give an optional reason for each answer. Questions 6, 7, and 8 ask for the answer that is NOT correct. 5 points each. For problems 10 through 15, give a reason to justify your answer: 4 points for reason, 1 point for correct answer. Do all 16 questions.

1. Which of the following is the most common way to send data from a view to the controller?  
   a. By setting an instance variable (with a @ prefix.)  
   b. By setting a local variable (without an @ prefix).  
   c. Through a params statement in the controller.  
   d. Through the session variable.
2. What does REST mean when referring to Ruby on Rails?  
   a. Using HTTP to implement the CRUD operations.  
   b. Using the DRY principle to reduce the number of lines of code.  
   c. Using a layout pages to increase performance.  
   d. Using cryptographic hashes to improve security.
3. Which line is correct?  
   a. **<%= f.text\_field\_tag :gender, '', :style -> ctrl %>**b. **<%= text\_field\_tag gender, '', style -> ‘ctrl’ %>**  
   c. **<%= text\_field :gender, '', :style -> ‘ctrl’ %>**  
   d. **<%= text\_field\_tag :gender, '', :style => ‘ctrl’ %>**
4. Which line is correct?  
   a. **<% form\_for 'page1' do |f| %>**  
   b. **<% form\_tag :action => 'page1' do %>**c. **<% form\_tag 'page1' do %>**d. **<% form\_tag :url => { :action => 'page1' } do %>**
5. Convert this rgb color specification to a hex color code:  
   **rgb(189,183,107)** (the color is DarkKhaki)  
   a. **#8CC76C** b. **#BDB76B**  c. **#CEC7B7** d. **#E8B7C7**
6. In this CSS style, what are the text color and the background color?  
   **body { color: #FFFF00; background\_color: #008080; }**a. fuchsia; olive b. yellow; olive c. purple; yellow d. yellow; teal
7. Which of these colors is **NOT** web-safe?  
   a. **#000000** b. **#33CCFF**  c. **#C0C0C0** d. **#FF6600**
8. What does this CSS style do?  
   **img { float: left; }**  
   a. Causes the image to float above any other images to the left.  
   b. Causes the image to float to the top of the page.   
   c. Causes the paragraph text to flow around the left of the of the image.  
   d. Causes the paragraph text to flow around the right of the image.
9. Which of the following is **NOT** a use for base 64 representation?  
   a. Binary data in XML files. b. Cryptographic hashes.  
   c. CSS color codes. d. Rails authenticity token.
10. \*What is the output?   
    **a = [6, 4, 9, 6, 8, 5, 7, 6]  
    n = 4  
    m = 2  
    puts a[n + m]**a. 5 b. 6 c. 7 d. 8
11. \*What is the output?   
    **a = [6, 3, 5]  
    b = a.collect do |i|  
     i \* i + 2  
    end**a. **[36, 9, 25]** b. **[38, 11, 27]** c. **[48, 15, 35]** d. **[25]**
12. \*Which line is correct?  
    a. **hash = [ 'c34' -> 'dog', 'c34' -> 'cat' ]**b. **hash = { 'c34' => 'dog', 'c34' => 'cat' }**c. **hash = { /c34/ -> 'dog', 'c34' -> 'cat' }**   
    d. **hash = { /c34/ => 'dog', /c34/ -> 'cat' }**
13. \*What is the output?   
    **s = ''  
    (1..4).each do |value|  
     s = value.to\_s + s  
    end**a. 1111 b. 1234 b. 4321 d. 4444
14. \*Which of the following character strings is NOT validated by this regular expression?  
    **/\A(-?\d+)|(\(d+\))\Z/**  
    a. 34563543 b. -342 c. (342) d. (-3827)

1. \*Convert this base 64 value to a character string: **ZW5k**
2. What is a cryptographic hash?  
   a. A one-way encryption of plaintext. b. An nonencoded password.  
   c. A sequence of numbers. d. A pair of hex color codes.