

IT 232 – Early Final Exam

November 6, 2015

Part A: Multiple Choice Questions. Circle the correct answer. For each question, give an optional reason: 4 points for reason, 1 point for correct answer. If answer is correct, the reason will not be considered. 5 points each.

- Which of the following is used to start a comment in an HTML or ERB source code file?
 a. `/*` b. `//` c. `<!--` d. `<%=`
- Which of these is **NOT** an example of being DRY in the Rails sense?
 a. Putting CSS styles in an external style sheet that is linked to several views.
 b. Putting content shown on several views in a layout page.
 c. Putting code for a form shown on more than one view in a form file such as `_form.html.erb`.
 d. Putting the code for each view in a separate file.
- A scaffold based Rails project is named Store with model StockItem. Which URL helper method call takes you to the **edit** view in the **stock_items** controller?
 a. `edit_stock_item_path` b. `edit_stock_item_path(@stock_item)`
 c. `edit_stock_items_path` d. `edit_stock_items_path(@stock_item)`
- Which of the following causes HTML or ERB code to be embedded within in a view?
 a. **embed** b. **insert** c. **render** d. **yield**
- Which of the following HTTP methods requires the data to be embedded in the URL when sending a form back to the server? a. **GET** b. **HEAD** c. **POST** d. **PUT**
- Which ActiveRecord command returns an array of Comment objects, sorted in order of most recent to least recent?
 a. `Comment.order(created_at: 'desc').all`
 b. `Comment.order(:created_at => 'descending').all`
 c. `Comment.sort(time: 'descending').all`
 d. `Comment.sort(time: descending).all`
- What does the jQuery selector `$("#p")` select?
 a. All elements with class="p". b. All elements with **id**="p". c. All paragraph tags. d. The first paragraph.
- Which CoffeeScript symbol is used to define an event handler function?
 a. `->` b. `-->` c. `=>` d. `<=>`

9. The routes for the **Patient** model are nested within the routes for the **Doctor** model as defined by

```
Rails.application.routes.draw do
  resources :doctors do
    resources :patients
  end
end
```

Which URL is used to access the patient with id=25 that has the doctor with id=7?

- `http://localhost:3000/doctors/patients`
 - `http://localhost:3000/doctors/patients/25`
 - `http://localhost:3000/doctors/7/patients/25`
 - `http://localhost:3000/patients/25/doctors/7`
10. For a class, defining which Ruby operator allows Ruby to sort an array of objects from this class with the **Array** **sort** method?
- `<<`
 - `<=`
 - `+=`
 - `<=>`

Part B. Make corrections to this Rails project. 30 points total. On the input view, this project inputs the name, currency code, and original amount. When the submit button is clicked, the application uses the currency code to look up the exchange rate in the database table currencies. It then multiplies the original amount by the exchange rate to obtain the converted amount, which is displayed on the display view, together with the name, currency code, and original amount. Correct the errors directly on the exam page – do not recopy. There are 3 to 4 errors per source code file.

--- CurrencyExchange/app/views/convert/input.html.erb -----

```
<% form_tag display_convert_path %>

<p>Name:<br />

<%= text_field_tag :name, '' class: 'ctrl' %></p>

<p>Currency Code:<br>
<%= select_tag(:currency_code,
  options_for_select(      <!-- This line is correct. -->
    [['Canadian Dollars',   'CAD'],
    ['European Union Euro', 'EUR'],
    ['British Pounds',      'GBP'],
    ['United States Dollars', 'USD'],
    ['Mexican Peso',        'MXP']],
  class: 'ctrl') %></p>

<p>Original Amount:<br />
<%= text_field_tag :orig_amt, '0.00',
  class: 'ctrl' %></p>

<p><%= submit_tag 'Convert Currency' %></p>
<% end %>
```

```
--- CurrencyExchange/app/views/convert/display.html.erb -----
```

```
<p>Currency Code: <%= @currency_code %></p>
<p>Original Amount: <%= @orig_amt %><p>
<p>Converted Amount: <%= converted_amount %></p>
```

```
--- CurrencyExchange/app/assets/stylesheets/display.scss -----
```

```
h2, input, select, body
{ background-color: #eeeeee;
  color: #eeeeee
  font-family: Verdana; }
```

```
input select
{ width: 2in; }
```

```
--- CurrencyExchange/app/views/layouts/application.html.erb -----
```

```
<DOCTYPE html>
<html>
<head>                                <!-- The head section is correct. -->
  <title><%= title %></title>
  <%= stylesheet_link_tag    'application', media: 'all',
    'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>

<h2><%= title %></h2>
<% yield %>

</html>
</body>
```

```
--- CurrencyExchange/db/seeds.db -----
```

```
Currency.create(code: 'CAD', rate: 1.330)
Currency.create(code: 'EUR', rate: 1.074)
Currencies.create(code: 'GBP', rate: 1.505)
Currencies.create(code: 'USD', rate: 1.000)
Currencies.create(:code => 'MXP', rate: 0.0595)
```

```
--- CurrencyExchange/app/controllers/display_controller.rb -----
```

```
class ConvertController < ApplicationController
  def input
    @title = Input Page
  end

  def display
    @title = "Display Page"
    @name = params[:name]
    @currency_code = params[:currency_code]
    @orig_amt = params[:orig_amt]
    exchange_rate = Currency.where(
      code: @currency_code).first.rate
    @converted_amt = @orig_amt * exchange_rate
  end
end
```

```
--- CurrencyExchange/app/assets/javascripts/display.js -----
```

```
//= require jquery
//= require jquery_ujs
//= require turbolinks
//= require_tree .

$('document').ready( function( ) {
  $('.ctrl').focus( function( ) {
    $(this).css('backgroundColor', 'maroon');
    $(this).css('color', yellow);
  });
  $('.ctrl').blur( function( ) {
    $(this).css('background-color', '#eeeeee');
    $(this).css('color', '#600000');
  });
});
```

```
--- CurrencyExchange/db/migrate/20151107192430_create_currencies.rb -----
```

```
class CreateCurrencies < ActiveRecord::Migration
  def change
    create_table :currencies |t| do
      t.string :code
      t.string :rate
      t.timestamps null: false
    end
  end
end
```

Part C. Modify a Rails scaffold project. The code on pages 7 to 25 is from the Depot Rails scaffold project. Make the source code modifications as specified in items 1 through 10 directly on the source code pages. Even if you cannot correctly change the source code to make the requested modification for a question, explain what you would do in general for that question. 5 points each.

1. A customer has many shopping carts, although only one is active at a given time. Both shopping carts and products have many line items. Implement these one-to-many relationships.
2. On the top of the **Customer** show view, add a **<h2>** header that displays the name of the customer something like this:
Customer: Jason Robinson
 Of course “Jason Robinson” should be replaced by the actual customer name from the database.
3. Change the text on the **Product index** view to make this view more readable and user friendly.
4. On the **LineItem index** and **show** views, replace the **shopping_cart_id** and the **product_id** with the customer name and product catalog number, respectively.
5. On the **LineItem index** and **show** views, add the extended price as a display item. The extended price of an item is the price for one item, multiplied by the quantity. (**quantity** is a field in the **line_items** table.)
6. On the **ShoppingCart** input form that is used on both the new and edit views, replace the number field control **:customer_id** with a dropdown menu (select control) that allows the user to select the customer name, but send the **customer_id** back to the controller when the form is submitted. You will need to define a two-dimensional options array for which the first column contains the customer name but the second column contains the customer id.
7. Suppose that the **app/assets/images** folder contains photos of each item in the products table. For example, if the catalog number (**catalog_num**) of the product is **C34574-31**, the image file of the photo for this product is **C34574-31.png**. Display this photo on the **Product** show view.
8. On the **customer show** view, display the line items in the most recent shopping cart item as determined by the **ShoppingCart created_at** field.
9. On the layout page, add hyperlinks to the index views of the **Customer**, **Product**, **ShoppingCart**, and **LineItem** views. Since they are on the layout page, these hyperlinks will appear on all views in the project. Why would the **link_to_unless_current** helper method be appropriate in this situation?
10. In general, explain what further changes you might make to the layout and styles of the views in this project so that the project does not look like an “out-of-the-box” Rails scaffold project.

Table of Contents for the Depot Project

app/models/customer.rb	Page 7
app/models/product.rb	Page 7
app/models/shopping_cart.rb	Page 7
app/models/line_item.rb	Page 7
 app/controllers/customers_controller.rb	 Pages 8, 9
app/views/customers/index.html.erb	Page 10
app/views/customers/new.html.erb	Page 10
app/views/customers/edit.html.erb	Page 10
app/views/customers/show.html.erb	Page 11
app/views/customers/_form.html.erb	Page 11
 app/controllers/products_controller.rb	 Pages 12, 13
app/views/products/index.html.erb	Page 14
app/views/products/new.html.erb	Page 14
app/views/products/edit.html.erb	Page 14
app/views/products/show.html.erb	Page 15
app/views/products/_form.html.erb	Page 15
 app/controllers/shopping_carts_controller.rb	 Pages 16, 17
app/views/shopping_carts/index.html.erb	Page 18
app/views/shopping_carts/new.html.erb	Page 18
app/views/shopping_carts/edit.html.erb	Page 18
app/views/shopping_carts/show.html.erb	Page 19
app/views/shopping_carts/_form.html.erb	Page 19
 app/controllers/line_items_controller.rb	 Pages 20, 21
app/views/line_items/index.html.erb	Page 22
app/views/line_items/new.html.erb	Page 22
app/views/line_items/edit.html.erb	Page 22
app/views/line_items/show.html.erb	Page 23
app/views/line_items/_form.html.erb	Page 23
 app/views/layouts/application.html.erb	 Page 24
app/assets/stylesheets/customers.scss	Page 25
app/assets/stylesheets/products.scss	Page 25
app/assets/stylesheets/shopping_carts.scss	Page 25
app/assets/stylesheets/line_items.scss	Page 25

---- Depot/app/models/customer.rb -----

```
class Customer < ActiveRecord::Base
```

```
end
```

---- Depot/app/models/product.rb -----

```
class Product < ActiveRecord::Base
```

```
end
```

---- Depot/app/models/shopping_cart.rb -----

```
class ShoppingCart < ActiveRecord::Base
```

```
end
```

---- Depot/app/models/line-item.rb -----

```
class LineItem < ActiveRecord::Base
```

```
end
```

---- Depot/app/controllers/customers_controller.rb ----

```
class CustomersController < ApplicationController
  before_action :set_customer, only: [:show, :edit, :update, :destroy]

  # GET /customers
  # GET /customers.json
  def index
    @customers = Customer.all
  end

  # GET /customers/1
  # GET /customers/1.json
  def show
  end

  # GET /customers/new
  def new
    @customer = Customer.new
  end

  # GET /customers/1/edit
  def edit
  end

  # POST /customers
  # POST /customers.json
  def create
    @customer = Customer.new(customer_params)

    respond_to do |format|
      if @customer.save
        format.html { redirect_to @customer,
          notice: 'Customer was successfully created.' }
        format.json { render :show, status: :created, location: @customer }
      else
        format.html { render :new }
        format.json { render json: @customer.errors,
          status: :unprocessable_entity }
      end
    end
  end
end
```


---- Depot/app/controllers/customers_controller.rb ----- Continued.

```
# PATCH/PUT /customers/1
# PATCH/PUT /customers/1.json
def update
  respond_to do |format|
    if @customer.update(customer_params)
      format.html { redirect_to @customer,
        notice: 'Customer was successfully updated.' }
      format.json { render :show, status: :ok, location: @customer }
    else
      format.html { render :edit }
      format.json { render json: @customer.errors,
        status: :unprocessable_entity }
    end
  end
end

# DELETE /customers/1
# DELETE /customers/1.json
def destroy
  @customer.destroy
  respond_to do |format|
    format.html { redirect_to customers_url,
      notice: 'Customer was successfully destroyed.' }
    format.json { head :no_content }
  end
end

private
# Use callbacks to share common setup or constraints between actions.
def set_customer
  @customer = Customer.find(params[:id])
end

# Never trust parameters from the scary internet,
# only allow the white list through.
def customer_params
  params.require(:customer).permit(:name, :phone)
end
end
```

```
---- Depot/app/views/customers/index.html.erb -----
```

```
<p id="notice"><%= notice %></p>
```

```
<h1>Listing Customers</h1>
```

```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Phone</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @customers.each do |customer| %>
      <tr>
        <td><%= customer.name %></td>
        <td><%= customer.phone %></td>
        <td><%= link_to 'Show', customer %></td>
        <td><%= link_to 'Edit', edit_customer_path(customer) %></td>
        <td><%= link_to 'Destroy', customer, method: :delete,
          data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>
```

```
<br>
```

```
<%= link_to 'New Customer', new_customer_path %>
```

```
---- Depot/app/views/customers/new.html.erb -----
```

```
<h1>New Customer</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Back', customers_path %>
```

```
---- Depot/app/views/customers/edit.html.erb -----
```

```
<h1>Editing Customer</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Show', @customer %> |
```

```
<%= link_to 'Back', customers_path %>
```

---- Depot/app/views/customers/show.html.erb -----

```
<p id="notice"><%= notice %></p>

<p>
  <strong>Name:</strong>
  <%= @customer.name %>
</p>

<p>
  <strong>Phone:</strong>
  <%= @customer.phone %>
</p>

<%= link_to 'Edit', edit_customer_path(@customer) %> |
<%= link_to 'Back', customers_path %>
```

---- Depot/app/views/customers/_form.html.erb -----

```
<%= form_for(@customer) do |f| %>
  <% if @customer.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@customer.errors.count, "error") %>
        prohibited this customer from being saved:</h2>

      <ul>
        <% @customer.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :name %><br>
    <%= f.text_field :name %>
  </div>
  <div class="field">
    <%= f.label :phone %><br>
    <%= f.text_field :phone %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>

<% end %>
```

---- Depot/app/controllers/products_controller.rb -----

```
class ProductsController < ApplicationController
  before_action :set_product, only: [:show, :edit, :update, :destroy]

  # GET /products
  # GET /products.json
  def index
    @products = Product.all
  end

  # GET /products/1
  # GET /products/1.json
  def show
  end

  # GET /products/new
  def new
    @product = Product.new
  end

  # GET /products/1/edit
  def edit
  end

  # POST /products
  # POST /products.json
  def create
    @product = Product.new(product_params)

    respond_to do |format|
      if @product.save
        format.html { redirect_to @product,
          notice: 'Product was successfully created.' }
        format.json { render :show, status: :created, location: @product }
      else
        format.html { render :new }
        format.json { render json: @product.errors, status:
          :unprocessable_entity }
      end
    end
  end
end
```

---- Depot/app/controllers/products_controller.rb ----- Continued

```
# PATCH/PUT /products/1
# PATCH/PUT /products/1.json
def update
  respond_to do |format|
    if @product.update(product_params)
      format.html { redirect_to @product,
        notice: 'Product was successfully updated.' }
      format.json { render :show, status: :ok, location: @product }
    else
      format.html { render :edit }
      format.json { render json: @product.errors,
        status: :unprocessable_entity }
    end
  end
end

# DELETE /products/1
# DELETE /products/1.json
def destroy
  @product.destroy
  respond_to do |format|
    format.html { redirect_to products_url,
      notice: 'Product was successfully destroyed.' }
    format.json { head :no_content }
  end
end

private
# Use callbacks to share common setup or constraints between actions.
def set_product
  @product = Product.find(params[:id])
end

# Never trust parameters from the scary internet,
# only allow the white list through.
def product_params
  params.require(:product).permit(:catalog_num, :description, :price)
end
end
```

```
---- Depot/app/views/products/index.html.erb -----
```

```
<p id="notice"><%= notice %></p>
```

```
<h1>Listing Products</h1>
```

```
<table>
  <thead>
    <tr>
      <th>Catalog num</th>
      <th>Description</th>
      <th>Price</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @products.each do |product| %>
      <tr>
        <td><%= product.catalog_num %></td>
        <td><%= product.description %></td>
        <td><%= product.price %></td>
        <td><%= link_to 'Show', product %></td>
        <td><%= link_to 'Edit', edit_product_path(product) %></td>
        <td><%= link_to 'Destroy', product, method: :delete,
          data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>
<br>
```

```
<%= link_to 'New Product', new_product_path %>
```

```
---- Depot/app/views/products/new.html.erb -----
```

```
<h1>New Product</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Back', products_path %>
```

```
---- Depot/app/views/products/edit.html.erb -----
```

```
<h1>Editing Product</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Show', @product %> |
```

```
<%= link_to 'Back', products_path %>
```

---- Depot/app/views/products/show.html.erb -----

```
<p id="notice"><%= notice %></p>
```

```
<p>
  <strong>Catalog num:</strong>
  <%= @product.catalog_num %>
</p>
```

```
<p>
  <strong>Description:</strong>
  <%= @product.description %>
</p>
```

```
<p>
  <strong>Price:</strong>
  <%= @product.price %>
</p>
```

```
<%= link_to 'Edit', edit_product_path(@product) %> |
<%= link_to 'Back', products_path %>
```

---- Depot/app/views/products/_form.html.erb -----

```
<%= form_for(@product) do |f| %>
  <% if @product.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@product.errors.count, "error") %>
        prohibited this product from being saved:</h2>
      <ul>
        <% @product.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>
  <div class="field">
    <%= f.label :catalog_num %><br>
    <%= f.text_field :catalog_num %>
  </div>
  <div class="field">
    <%= f.label :description %><br>
    <%= f.text_area :description %>
  </div>
  <div class="field">
    <%= f.label :price %><br>
    <%= f.text_field :price %>
  </div>
```

```

<div class="actions">
  <%= f.submit %>
</div>
<% end %>

```

```

---- Depot/controllers/shopping_carts_controller.rb ----

```

```

class ShoppingCartsController < ApplicationController
  before_action :set_shopping_cart, only: [:show, :edit, :update, :destroy]

  # GET /shopping_carts
  # GET /shopping_carts.json
  def index
    @shopping_carts = ShoppingCart.all
  end

  # GET /shopping_carts/1
  # GET /shopping_carts/1.json
  def show
  end

  # GET /shopping_carts/new
  def new
    @shopping_cart = ShoppingCart.new
  end

  # GET /shopping_carts/1/edit
  def edit
  end

  # POST /shopping_carts
  # POST /shopping_carts.json
  def create
    @shopping_cart = ShoppingCart.new(shopping_cart_params)
    respond_to do |format|
      if @shopping_cart.save
        format.html { redirect_to @shopping_cart,
          notice: 'Shopping cart was successfully created.' }
        format.json { render :show, status: :created, location: @shopping_cart }
      else
        format.html { render :new }
        format.json { render json: @shopping_cart.errors,
          status: :unprocessable_entity }
      end
    end
  end
end

```


---- Depot/controllers/shopping_carts_controller.rb ---- Continued

```
# PATCH/PUT /shopping_carts/1
# PATCH/PUT /shopping_carts/1.json
def update
  respond_to do |format|
    if @shopping_cart.update(shopping_cart_params)
      format.html { redirect_to @shopping_cart,
        notice: 'Shopping cart was successfully updated.' }
      format.json { render :show, status: :ok, location: @shopping_cart }
    else
      format.html { render :edit }
      format.json { render json: @shopping_cart.errors,
        status: :unprocessable_entity }
    end
  end
end

# DELETE /shopping_carts/1
# DELETE /shopping_carts/1.json
def destroy
  @shopping_cart.destroy
  respond_to do |format|
    format.html { redirect_to shopping_carts_url,
      notice: 'Shopping cart was successfully destroyed.' }
    format.json { head :no_content }
  end
end

private
# Use callbacks to share common setup or constraints between actions.
def set_shopping_cart
  @shopping_cart = ShoppingCart.find(params[:id])
end

# Never trust parameters from the scary internet, only allow
# the white list through.
def shopping_cart_params
  params.require(:shopping_cart).permit(:customer_id)
end
end
```

```
---- Depot/app/views/shopping_carts/index.html.erb ----
```

```
<p id="notice"><%= notice %></p>
```

```
<h1>Listing Shopping Carts</h1>
```

```
<table>
```

```
  <thead>
```

```
    <tr>
```

```
      <th>Customer</th>
```

```
      <th colspan="3"></th>
```

```
    </tr>
```

```
  </thead>
```

```
  <tbody>
```

```
    <% @shopping_carts.each do |shopping_cart| %>
```

```
      <tr>
```

```
        <td><%= shopping_cart.customer_id %></td>
```

```
        <td><%= link_to 'Show', shopping_cart %></td>
```

```
        <td><%= link_to 'Edit', edit_shopping_cart_path(shopping_cart) %></td>
```

```
        <td><%= link_to 'Destroy', shopping_cart,
          method: :delete, data: { confirm: 'Are you sure?' } %></td>
```

```
      </tr>
```

```
    <% end %>
```

```
  </tbody>
```

```
</table>
```

```
<br>
```

```
<%= link_to 'New Shopping cart', new_shopping_cart_path %>
```

```
---- Depot/app/views/shopping_carts/new.html.erb -----
```

```
<h1>New Shopping Cart</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Back', shopping_carts_path %>
```

```
---- Depot/app/views/shopping_carts/edit.html.erb -----
```

```
<h1>Editing Shopping Cart</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Show', @shopping_cart %> |
```

```
<%= link_to 'Back', shopping_carts_path %>
```

---- Depot/app/views/shopping_carts/show.html.erb -----

```
<p id="notice"><%= notice %></p>

<p>
  <strong>Customer:</strong>
  <%= @shopping_cart.customer_id %>
</p>

<%= link_to 'Edit', edit_shopping_cart_path(@shopping_cart) %> |
<%= link_to 'Back', shopping_carts_path %>
```

---- Depot/app/views/shopping_carts/_form.html.erb -----

```
<%= form_for(@shopping_cart) do |f| %>
  <% if @shopping_cart.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@shopping_cart.errors.count, "error") %>
        prohibited this shopping_cart from being saved:</h2>
      <ul>
        <% @shopping_cart.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :customer_id %><br>
    <%= f.number_field :customer_id %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

```
---- Depot/app/controllers/line_items_controller.rb ----
```

```
class LineItemsController < ApplicationController
  before_action :set_line_item, only: [:show, :edit, :update, :destroy]

  # GET /line_items
  # GET /line_items.json
  def index
    @line_items = LineItem.all
  end

  # GET /line_items/1
  # GET /line_items/1.json
  def show
  end

  # GET /line_items/new
  def new
    @line_item = LineItem.new
  end

  # GET /line_items/1/edit
  def edit
  end

  # POST /line_items
  # POST /line_items.json
  def create
    @line_item = LineItem.new(line_item_params)

    respond_to do |format|
      if @line_item.save
        format.html { redirect_to @line_item,
          notice: 'Line item was successfully created.' }
        format.json { render :show, status: :created, location: @line_item }
      else
        format.html { render :new }
        format.json { render json: @line_item.errors,
          status: :unprocessable_entity }
      end
    end
  end
end
```

---- Depot/app/controllers/line_items_controller.rb ---- (Continued)

```
# PATCH/PUT /line_items/1
# PATCH/PUT /line_items/1.json
def update
  respond_to do |format|
    if @line_item.update(line_item_params)
      format.html { redirect_to @line_item,
        notice: 'Line item was successfully updated.' }
      format.json { render :show, status: :ok, location: @line_item }
    else
      format.html { render :edit }
      format.json { render json: @line_item.errors,
        status: :unprocessable_entity }
    end
  end
end

# DELETE /line_items/1
# DELETE /line_items/1.json
def destroy
  @line_item.destroy
  respond_to do |format|
    format.html { redirect_to line_items_url,
      notice: 'Line item was successfully destroyed.' }
    format.json { head :no_content }
  end
end

private
# Use callbacks to share common setup or constraints between actions.
def set_line_item
  @line_item = LineItem.find(params[:id])
end

# Never trust parameters from the scary internet, only allow
# the white list through.
def line_item_params
  params.require(:line_item).permit(:quantity, :shopping_cart_id,
    :product_id)
end
end
```

```
---- Depot/app/views/line_items/index.html.erb ----
```

```
<p id="notice"><%= notice %></p>
```

```
<h1>Listing Line Items</h1>
```

```
<table>
  <thead>
    <tr>
      <th>Quantity</th>
      <th>Shopping cart</th>
      <th>Product</th>
      <th colspan="3"></th>
    </tr>
  </thead>

  <tbody>
    <% @line_items.each do |line_item| %>
      <tr>
        <td><%= line_item.quantity %></td>
        <td><%= line_item.shopping_cart_id %></td>
        <td><%= line_item.product_id %></td>
        <td><%= link_to 'Show', line_item %></td>
        <td><%= link_to 'Edit', edit_line_item_path(line_item) %></td>
        <td><%= link_to 'Destroy', line_item, method: :delete,
          data: { confirm: 'Are you sure?' } %></td>
      </tr>
    <% end %>
  </tbody>
</table>
```

```
<br>
```

```
<%= link_to 'New Line item', new_line_item_path %>
```

```
---- Depot/app/views/line_items/new.html.erb -----
```

```
<h1>New Line Item</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Back', line_items_path %>
```

```
---- Depot/app/views/line_items/edit.html.erb -----
```

```
<h1>Editing Line Item</h1>
```

```
<%= render 'form' %>
```

```
<%= link_to 'Show', @line_item %> |
```

```
<%= link_to 'Back', line_items_path %>
```

---- Depot/app/views/line_items/_form.html.erb ----

```
<%= form_for(@line_item) do |f| %>
  <% if @line_item.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@line_item.errors.count, "error") %>
        prohibited this line_item from being saved:</h2>

      <ul>
        <% @line_item.errors.full_messages.each do |message| %>
          <li><%= message %></li>
        <% end %>
      </ul>
    </div>
  <% end %>

  <div class="field">
    <%= f.label :quantity %><br>
    <%= f.number_field :quantity %>
  </div>
  <div class="field">
    <%= f.label :shopping_cart_id %><br>
    <%= f.number_field :shopping_cart_id %>
  </div>
  <div class="field">
    <%= f.label :product_id %><br>
    <%= f.number_field :product_id %>
  </div>
  <div class="actions">
    <%= f.submit %>
  </div>
<% end %>
```

---- Depot/app/views/line_items/show.html.erb -----

```
<p id="notice"><%= notice %></p>

<p>
  <strong>Quantity:</strong>
  <%= @line_item.quantity %>
</p>

<p>
  <strong>Shopping cart:</strong>
  <%= @line_item.shopping_cart_id %>
</p>
```

```
<p>
  <strong>Product:</strong>
  <%= @line_item.product_id %>
</p>
```

```
<%= link_to 'Edit', edit_line_item_path(@line_item) %> |
<%= link_to 'Back', line_items_path %>
```

```
----- Depot/app/views/layouts/application.html.erb -----
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Depot</title>
  <%= stylesheet_link_tag 'application', media: 'all',
    'data-turbolinks-track' => true %>
  <%= javascript_include_tag 'application', 'data-turbolinks-track' => true %>
  <%= csrf_meta_tags %>
</head>
<body>
```

```
<%= yield %>
```

```
</body>
</html>
```


---- Depot/app/assets/stylesheets/customers.scss -----

// Place all the styles related to the Customers controller here.
// They will automatically be included in application.css.
// You can use Sass (SCSS) here: <http://sass-lang.com/>

---- Depot/app/assets/stylesheets/products.scss -----

---- Depot/app/assets/stylesheets/shopping_carts.scss --

---- Depot/app/assets/stylesheets/line_items.scss -----

Part D. Short Essay. Answer **one** of the three following questions. For full credit, write in complete sentences and paragraphs. 10 points.

1. What does REST mean? How is it related to CRUD?
2. What is the difference between **form_tag** and **form_for** for defining a form?
3. What is a Rails session variable and how is it useful?