COMP 4081 **Exam 1** Fall 2014

Name:		,
	Last name	First name

Rules:

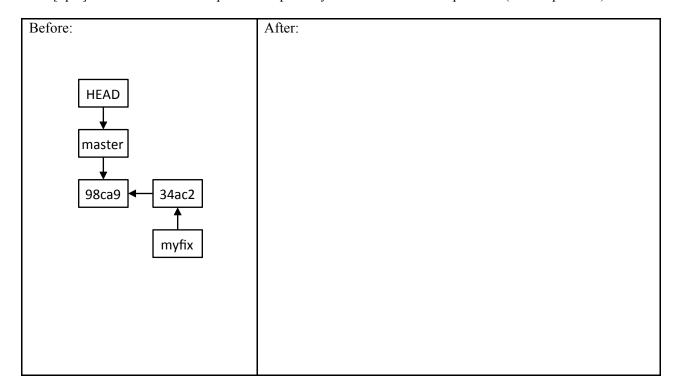
- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.
- <u>WEIRD!</u> Do not write on the backs of pages. If you need more pages, ask me for some.

Reminders:

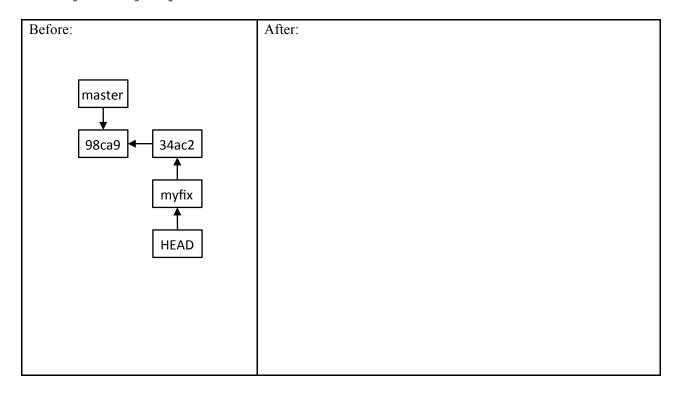
- Verify that you have all pages.
- Don't forget to write your name.
- Read each question <u>carefully</u>.
- Don't forget to answer <u>every</u> question.

1.	[3pts]	The Git clone command does which of the following?
	a.	Creates a working directory
	b.	Makes a local copy of the repository
	c.	Commits a new branch
	d.	a and b
	e.	a, b, and c
2.		Which Git command changes where the HEAD pointer points and modifies the contents of the g directory?
	a.	checkout
	b.	merge
	c.	mv
	d.	pull
	e.	None of the above
3.	[3pts]	Which one of the following is <u>not</u> part of the data structure of a Git repository?
	a.	Body element
	b.	Branch pointer
	c.	Commit object
	d.	HEAD pointer
	e.	None of the above (i.e., they are all parts)

4. [8pts] Draw the state of the pictured repository after a Git **commit** operation (make up a hash).



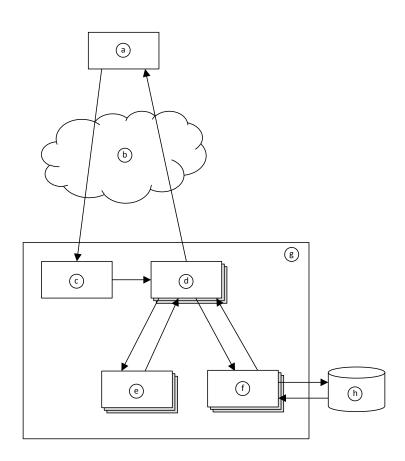
- 5. [8pts] Draw the state of the pictured repository after running the following commands.
 - \$ git checkout master
 - \$ git merge myfix



6. [8pts] Given the Rails MVC architectural diagram below, label each component.

a.	

- b. _____
- c. ____
- d. _____
- e. _____
- f. _____
- g. ____
- h.



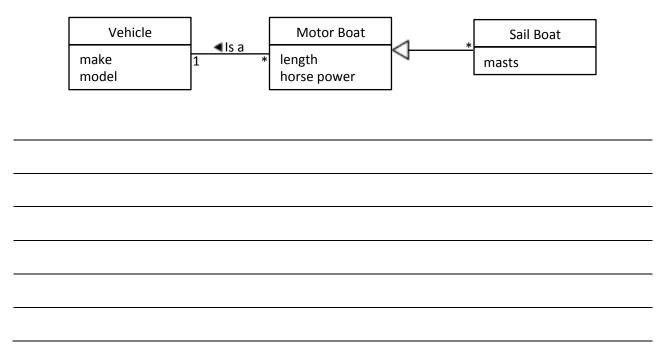
7.	[14pts] Figures 1–6 pertain to a rental-property web app. Write Ruby code that defines the show method in Figure 5, and write the ERB code that would produce the page depicted in Figure 2. Assume that a layout, application.html.erb, already exists, so your ERB needs only to include the main content being displayed. Your ERB must have the following types of HTML elements: p and strong .

8.		Why would it violate the SRP to move line 3 from RentalsController (Figure 5) into the ing of index.html.erb (Figure 6)?
9.	[3pts]	Which of the following is true of <i>exhaustive testing</i> ?
	a.	Generally infeasible in practice
	b.	Tests all possible inputs
	c.	Typically results in an intractably large set of test cases even for small programs
	d.	All of the above
	e.	None of the above
10	. [3pts]	Which type(s) of tests do you typically write when doing test-driven development?
	a.	Automated tests
	b.	Unit tests
	c.	Blackbox tests
	d.	All of the above
	e.	None of the above

11. [9pts] Following test-driven development (TDD), describe in plain English how you would go about adding a new method index2br to RentalsController along with an associated ERB. The purpose of this functionality would be to display an index of all rentals with 2 bedrooms. Describe the sequence of steps you would perform and all the tests you would write. Cover all aspects of TDD.

	12. [17pts] Draw a domain model in the form of a class diagram based on the following description. Model only things that are specifically described. Include all conceptual classes, attributes, associations, and generalization relationships mentioned. Label all associations and include all multiplicities. Do not model "the system."
	A system is needed for managing rental properties. Each rental property has a monthly rent price and square footage. Each rental is owned by a landlord with a name, a phone number, and an address. There are two types of rentals: home and business. Home rentals have a number of bedrooms and a number of bathrooms. Business rentals have a number of offices and a maximum occupancy. A rental may adjoin a number of other rental properties (like apartments in the same complex). If a rental property is currently rented, it will have an associated lease with the name of the renter, the renter's phone number, a start date, and an end date.
_	
_	
_	
_	
_	
_	

13. [3pts] Name three problems with this domain model (esp. with respect to class diagram notation).



- 14. [3pts] Which of the following is meant by software engineering process?
 - a. A structure imposed on the development of a software product; for example, developing iteratively and incorporating best practices might be ingredients in an SE process
 - b. Something developers use to accomplish a goal during a project; for example, Git is an SE process for configuration management
 - c. Something developers do to accomplish a goal during a project; for example, planning poker is an SE process for estimation
 - d. A running instance of a program; for example, a UNIX process is an SE process
 - e. None of the above
- 15. [3pts] An empirical process-control model iterates between
 - a. design and implementation
 - b. requirements gathering and design
 - c. user studies and testing
 - d. feedback and adaptation
 - e. None of the above

16. [9pts] Please answer the following 3 related questions:	
a. What often-false assumption does the waterfall software engineering process make?b. Why does this false assumption cause considerable problems for waterfall?	
c. How does iterative development overcome these problems?	
c. The does iterative development overcome these problems:	

Figures

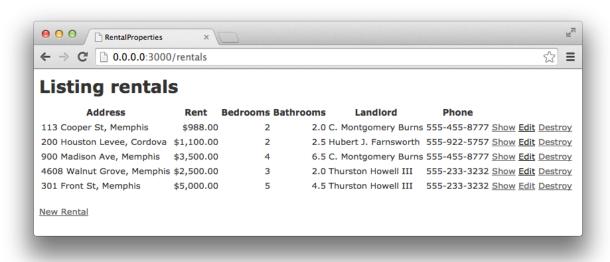


Figure 1. Index page for rental-property web app.

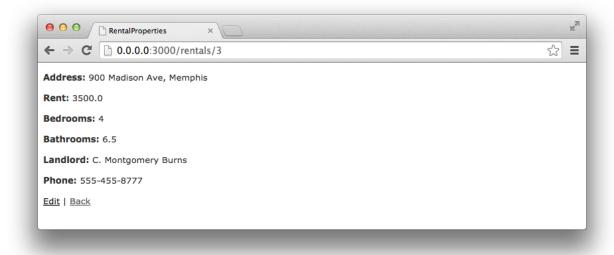


Figure 2. Show-rental page for rental-property web app.

```
$ rake routes
    Prefix Verb URI Pattern
                                              Controller#Action
                  /rentals(.:format)
   rentals GET
                                             rentals#index
           POST
                 /rentals(.:format)
                                             rentals#create
new_rental GET
                  /rentals/new(.:format)
                                              rentals#new
                  /rentals/:id/edit(.:format) rentals#edit
edit_rental GET
    rental GET
                  /rentals/:id(.:format)
                                             rentals#show
           PATCH /rentals/:id(.:format)
                                              rentals#update
                  /rentals/:id(.:format)
                                             rentals#update
           PUT
           DELETE /rentals/:id(.:format)
                                             rentals#destroy
```

Figure 3. Result of "rake routes" command for rental-property web app.

```
1
    # == Schema Information
2
3
    # Table name: rentals
 4
5
    # id
                                     not null, primary key
                   :integer
                   :string(255)
6
       address
 7
       rent
                   :decimal(, )
8
    # bedrooms
                   :integer
9
    # bathrooms :float
10
    # landlord
                   :string(255)
11
    # phone
                   :string(255)
    # created_at :datetime
12
13
       updated_at :datetime
14
15
16
    class Rental < ActiveRecord::Base</pre>
17
    end
```

Figure 4. Rental-property web app file: app/models/rental.rb

```
1 ▼ class RentalsController < ApplicationController
       def index
        @rentals = Rental.all
 3
 4
       end
 5
 6
       def show
       # YOUR ANSWER HERE
 7
 8
       end
 9
10
       def new
       @rental = Rental.new
11
12
       end
13
14
       def edit
15
       @rental = Rental.find(params[:id])
16
       end
17
18 ▼
       def create
         @rental = Rental.new(rental_params)
19
20 ▼
         respond_to do |format|
21 ▼
           if @rental.save
             format.html { redirect_to @rental, notice: 'Rental was successfully created.' }
22
23
             format.json { render action: 'show', status: :created, location: @rental }
24 ₩
           else
25
             format.html { render action: 'new' }
26
             format.json { render json: @rental.errors, status: :unprocessable_entity }
27
           end
28
         end
29
       end
       ... and so on ...
```

Figure 5. Rental-property web app file: app/controllers/rentals_controller.rb

```
1
   <h1>Listing rentals</h1>
2
3
   4
    <thead>
5
     6
       Address
7
      Rent
8
      Bedrooms
9
       Bathrooms
10
      Landlord
      Phone
11
12
      13
      > / th>
14
      15
     16
    </thead>
17
18
    19
     <% @rentals.each do |rental| %>
20
      21
        <%= rental.address %>
22
        <%= number_to_currency(rental.rent) %>
23
        <%= rental.bedrooms %>
        <%= rental.bathrooms %>
24
        <%= rental.landlord %>
25
26
        <%= rental.phone %>
        27
28
29
        <$ link_to 'Destroy', rental, method: :delete, data: { confirm: 'Are you sure?' } %>
        30
31
     <% end %>
32
    33
   34
35
   <br>
36
   <%= link_to 'New Rental', new_rental_path %>
```

Figure 6. Rental-property web app file: app/views/index.html.erb