University of Toronto

Faculty of Applied Science and Engineering

Midterm – November, 2016

CSC444 --- Software Engineering I

Examiner: Michael Stumm

Instructions: Please read carefully --- marks are deducted if not followed.

- Write you last name, first name, and student number in the fields below
- Write your name and student number at the top of every page of this exam.
- This exam has eleven (11) pages.
- No additional sheets are permitted.
- Do not remove any sheets from this exam.
- There are a total of 10 questions and the weight of each question is the same.
- All questions must be answered on these sheets.
- If any of the questions appear unclear or ambiguous to you, then make any assumptions you need, state them and answer the question that way.
- Please write clearly so we can read what you write, and please use proper English so we can understand what you write.
- The use of calculators or computers is not permitted.
- Length of exam: 50 minutes
- Please manage time carefully; consider answering the easy questions first.

Last Name: _____

First Name: _____

Student Number: _____

Do not write below this line:

1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*

Total:

Name: _____ Student Number_____

1. Warmup questions

a. What is the name of this course:

Answer:

b. Using three words or less, the lab assignment is getting you to implement what?

Answer:

c. Identify the other members in your group by name or email address:

d. Consider the following two Ruby methods:

```
def times_two(arg1);
  puts arg1 * 2;
end

def sum(arg1, arg2);
  puts arg1 + arg2;
end
```

What will be the result of each of the following lines of code:

times_two 5	
times_two(5)	
times_two (5)	
sum 1, 2	
sum(1, 2)	
sum (1, 2)	

2. HTML, Ruby, and ERB

Consider the following code fragment of a Rails view:

```
<h1>Listing products</h1>

<table>
<% @products.each do |product| %>
  <tr class="<%= cycle('list_line_odd', 'list_line_even') %>">
    <td>
      <%= image_tag(product.image_url, class: 'list_image') %>
    </td>
    <td class="list_description">
      <dl>
        <dt><%= product.title %></dt>
        <dd><%= truncate(strip_tags(product.description), length: 80) %>
        </dd>
      </dl>
    </td>
  </tr>
<% end %>
</table>
<br />
<%= link_to 'New product', new_product_path %>
```

In the code above:

a) underline all Ruby code;

b) box all Erb code;

c) circle all Javascript code; and

d) cross out all HTML code (~~like this~~)

Name: _____ Student Number_____

3. Regular Expressions

a. What regular expression matches all items in the first column but none in the second column:

pit	
spot	**pt**
spate	**Pot**
slap two	**peat**
respite	**part**

Answer:

b. True/False: are the two regular expressions equivalent?

		True/False
`(ab)*a`	**`a(ba)*`**	
`a?a*`	**`a*`**	
`a*b*`	**`(ab)*`**	
`(0|1)*`	**`0?1?`**	

c. Give an English description of the sets of strings generated by the following regular expression. For full credit, give a description that describes the set without just rewriting the regular expressions in English. For example, if the regular expression is `(a*b*)*`, a good answer would be "*all strings with 0 or more a's and b's in any sequence*". A not so good answer would be "*zero or more repetitions of* ….".

`(x|y)*x(x|y)`

Answer:

Cheatsheet:

^ beginning of input	$ end of input	\ escape character
. any char	* zero or more	+ one or more
? zero or one	{n} n times	{n:m} n -m times
a|b matches a or b	[abd] any char in set	[^abd] any char but those in set
\w matches any word char	\s any white space char	\d matches any digit
	(x) match x and remember the match	

4. Ruby Metaprogramming

a. Using Ruby, give an example of a method with no name and one argument. and how it might be used:

b. Consider the following code. It invokes a method called define_method that expects two arguments. We did not cover this specific method in class, but the method generates a new method definition.

```
['admin', 'marketer', 'sales'].each do |user_role|
    define_method "#{user_role}?" do
        role == user_role
    end
end
```

In the space below, provide the methods and their implementations that are created as a result of the above code:

5. Models, Controllers, or Views I?

Label each of the tasks below with "Model", "View", or "Controller" to indicate where that task would typically be implemented in a Web application using an MVC architecture.

Validate form data	
Make sure a user is logged in	
Return a "redirect" to the browser	
Define a `before_create` callback	
Define a filter	
Generate a new session token	
Invoke the `find_all_by_name` method	
Create a "salt" for a password	

6. Git Revision Control System

Given the setup of repositories we are using for our labs, enter "Yes", "No", or "Maybe" in each empty cell of the following table:

Operation	May Fail	Modifies local branch	Modifies remote branch
`git commit`			
`git pull`			
`git push`			
`git status`			
`git diff`			

7. Models, Controllers, or Views II?

For each of the following code fragments, identify whether the code fragment would be part of a model, a controller, a view or some combination thereof, by writing the appropriate words into the column on the right.

`before_action :set_only [:show, :edit, :update]`
 	
`def index`
`  @products = Product.all`
`end`
 	
`def show`
`end`	

`validates :image_url, allow_blank: true, format: {`
`  with:  %r{\.(gif|jpg|png)\Z}I,`
`  message: 'must be GIF, JPG, or PNG image.'`	

`<%= product.title %>`	

`def change`
`  create_table :products do |t|`
`  t.string :title`
`  t.text :description`
`  t.string :image_url`
`  t.decimal :price, precision: 8, scale: 2`
`  t.timestamps`
`  end`
`end`	

`respond_to do |format|`
`  if @line_item.save`
`  format.html {redirect_to @line_item.cart,`
`  notice: 'Success!'`
`  else`
`  format.html { render action: 'new' }`
`  end`
`end`	

`if user = User.authenticate(params[:username],`
`params[:password])`
`  # Save the user ID in the session`
`  session[:current_user_id] = user.id`
`  redirect_to root_url`
`  end`	

8. Routes

The Rails convention maps HTTP methods to certain controller methods, and those methods usually involve specific CRUD operations on models. Given the following CRUD operations: `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`, and assuming you are implementing a controller for the resource `:prods` and have generated the corresponding routes automatically, complete the following table:

HTTP Method	URI	Controller # Action	CRUD operation
		`prods#index`	
		`prods#new`	
		`prods#create`	
		`prods#edit`	
		`prods#update`	
		`prods#destroy`	

9. BDD and Cucumber

Assume that a pop-up window is used to login users for the application. In the space below, provide Cucumber scenarios associated with loging in a user from this pop-up window.

10. Wind-down questions

a. Explain how you can tell whether a Ruby variable is

i. an instance variable:

Answer:

ii. a global variable:

Answer:

iii. a class variable:

Answer:

b. Explain what is `rake` in Rails?

Answer:

c. How can you list all routes for an application?

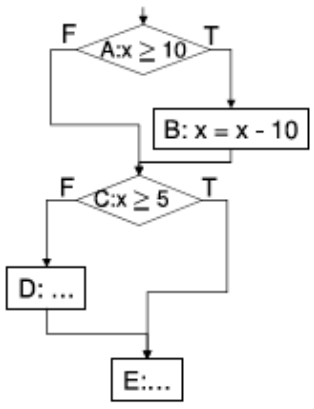
Answer:

d. White/Glass-box tests designed for one implementation are valid to use when testing another implementation: True or False?

e. Consider the following control-flow diagram, where A,B,C,D,E are names of code blocks. For each of the following code coverage criteria, find the ***minimum*** number of runs required to obtain the required coverage. Give an example of the initial value of x for each run, and write also the sequence of blocks being executed.

i. Statement coverage

Answer:



ii. Edge coverage

Answer:

iii. Path coverage

Answer: