

COMP 4081
Exam 2
Fall 2014

Name: _____,
Last name First name

Rules:

- No potty breaks.
- Turn off cell phones/devices.
- Closed book, closed note, closed neighbor.
- WEIRD! 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 carefully.
- Don't forget to answer every question.

1. [5pts] Fill in the 5 types of requirement that FURPS stands for.

F

U

R

P

S

2. [4pts] Consider the Seedpod system that we used in class. Reverse engineer one functional requirement for the system and write a user story for that requirement. Apply the templates and guidelines from lecture to ensure the US is high quality. You may omit the estimate and priority.

6. [7pts] Describe the process of iteration planning that we used in this course by writing 7 sentences. Create each sentence by filling in 3 blanks with the following words/phrases. Fill in *all* blanks.
- a. Blank #1: developer, customer
 - b. Blank #2: estimates, selects (for iteration), assigns (to developer), creates, prioritizes
 - c. Blank #3: tasks, user stories

_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

7. [4pts] Authentication is concerned with _____, whereas authorization is concerned with _____. (Fill in the blanks.)

- a. flagging the user if he/she misbehaves
- b. restricting what operations/data the user can access
- c. determining if the user is a hacker
- d. logging access violations
- e. determining who the user is

8. [3pts] How do you prevent packet-sniffing exploits?

- a. Escape packet text
- b. Scan for viruses
- c. Encrypt network communication with SSL
- d. Packet plugs
- e. None of the above

9. [3pts] Imagine a social networking web app (like Twitter) that allows users to post short blurbs of text. Which type of exploit might be carried out by posting text that contains malicious code?

- a. Cross-site scripting
- b. SQL injection
- c. Packet sniffing
- d. a and b
- e. a, b, and c

10. [7pts] Match the design pattern to the situation to which you should apply it.

- | | |
|-----------------------------------|--|
| Observer <input type="radio"/> | <input type="radio"/> Your Pac-Man program needs to listen for presses of the arrow keys and to update Pac-Man's position in the maze accordingly. |
| Builder <input type="radio"/> | <input type="radio"/> Your program has to create and configure some big, ugly record objects before inserting them into a database. |
| Adapter <input type="radio"/> | <input type="radio"/> Your GUI interface has many interrelated buttons and other widgets (e.g., such that when each button is pressed many other widgets must be updated). |
| Mediator <input type="radio"/> | <input type="radio"/> You want your application to save its state so that if it crashes, then it can auto-recover. |
| Memento <input type="radio"/> | <input type="radio"/> Your program must support switching among several different email libraries, but each one has a slightly different interface. |
| Interpreter <input type="radio"/> | <input type="radio"/> You want to let users create and run macros inside your application. |
| Facade <input type="radio"/> | <input type="radio"/> Sending an SMS message requires lots of big, ugly code, involving connection, message, and other objects. |

11. [11pts] Create a UML class diagram representing the point-of-sale model classes in Figure 1. Be sure to label all associations and association ends, and include all multiplicities. Don't include "id" attributes (objects have identity by default).

12. [11pts] Consider the following execution of a point-of-sale system with the model in Figure 1. Two users register: Alice Zed (azed@memphis.edu) and Bob Young (byoung@memphis.edu). Alice purchases the following things: 2 Bug Zappers (\$20 each) and 1 Garden Hose (\$12 each). Bob purchases the following things: 3 Bug Zappers and 1 Spider Spray (\$4 each). Later, Alice makes another purchase: 1 Spider Spray. Create an object diagram that depicts the model objects after this execution.

13. [10pts] Draw a control-flow graph for the function in Figure 2. Label each edge in the graph with an uppercase letter.

14. [7pts] Fill in the table below with a test suite that provides path coverage of the code from the previous question. In the covers column, list the relevant labeled edges in your CFG that each test case covers. Some cells in the table may be left blank.

Input			Expected Output	Covers
x	y	z		

15. [3pts] In system testing, should the software developers who wrote the code perform the testing? Explain your answer.

```
1 # id      :integer      not null, primary key
2 # name    :string
3 # email   :string
4 class User < ActiveRecord::Base
5   has_many :sales
6 end
```

```
1 # id      :integer      not null, primary key
2 class Sale < ActiveRecord::Base
3   belongs_to :user
4   has_many :line_items
5 end
```

```
1 # id      :integer      not null, primary key
2 # quantity :integer
3 class LineItem < ActiveRecord::Base
4   belongs_to :sale
5   belongs_to :item_description
6 end
```

```
1 # id      :integer      not null, primary key
2 # description :string
3 # price    :integer
4 class ItemDescription < ActiveRecord::Base
5   has_many :line_items
6 end
```

Figure 1. Model classes for a point-of-sale system.

```
def min_of_three(x, y, z)
  if x < y then
    if x < z then
      return x
    else
      return z
    end
  else
    if y < z then
      return y
    else
      return z
    end
  end
end
```

Figure 2. Sample function.