COMP/EECE 7012

Exam 2

Spring 2017

Name: Solutions ,
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. [8pts] For each piece of text below, place an "I" and/or a "W" next to it if it corresponds to *Iterative* and/or *Waterfall* development process, respectively.

__I,W__ A structure imposed on the development of a software product

__I__ Development of a system through repeated cycles and in smaller portions at a time, allowing software developers to take advantage of what was learned during development of earlier parts or versions of the system

__W__ Development of a system whereby progress is seen as flowing steadily downwards through the phases of conception, analysis, design, construction, testing, production, and maintenance, and wherein one should move to a phase only when its preceding phase is reviewed and verified

__W__ System defects are often discovered late in the development process

__I__ Good if your project has unstable requirements (i.e., that are prone to change)

__W__ Falsely assumes that requirements can be known from the start

__W__ Can lead to "analysis paralysis" wherein a considerable investment of time and effort is put into a project before any code is written

__I__ Has an "empirical" process control model

2. [6pts] Think of *Twitter*. Reverse engineer one user story that records a requirement for the system. You must apply the templates described in class, and your US must have the other attributes of good user stories, which we discussed in class. (You may omit the US's estimate and priority.)

Many possible answers. Here are the templates:

Title: <verb> <noun>

Description: As a <who>, I want a <what> <why>.

Here are some key attributes:

- Describe one thing
- Use customer's language
- No be long essay
- Not use technical terms

3. [4pts] Describe two things that are wrong with this user story for the *Piazza* system we used in class.

> jQuery Rich Text Editor
> The create-post page must have a jQuery rich text editor widget for editing the details text of the post.

(1) Mentions specific implementation technology, jQuery.

(2) Uses technical jargon, jQuery, that the customer might not understand.

(3) Mentions specific features of the user-interface design, pages and rich text editor widget.

4. [2pts] T or F? In the agile development process taught in class, the development team estimates each user story and decides the priority for each story.

   a) True

   [✓ b)] False – Customer decides priority

5. [2pts] T or F? The smaller the estimate, the more likely it is to be accurate.

   [✓ a)] True

   b) False

6. [2pts] T or F? Planning poker uses the "wisdom of the crowd" to estimate how long it will take to implement user stories.

   [✓ a)] True

   b) False

7. [2pts] What is *exhaustive testing*, and why don't people do it in practice?

Exhaustive testing is where you have a test case for every possible input.
People don't do it because, even for small inputs, it is too computationally intensive (e.g., takes too long) to run all the tests.

8. [2pts] For each piece of text below, place a "B" or a "W" next to it if it corresponds to *Black-Box* or *White-Box* testing, respectively.

__W__ Tests based on the implementation of a component

__B__ Tests based only on the interface of a component

__B__ Tests focus on boundary cases

__W__ Tests aim to achieve particular levels of code-coverage

9. [2pts] For each piece of text below, place a "U" and/or "I" and/or "S" next to it if it corresponds to *Unit* and/or *Integration* and/or *System* tests, respectively.

__I,S__ May have non-determinism

__U__ Should not perform I/O

__S__ Tests the whole system

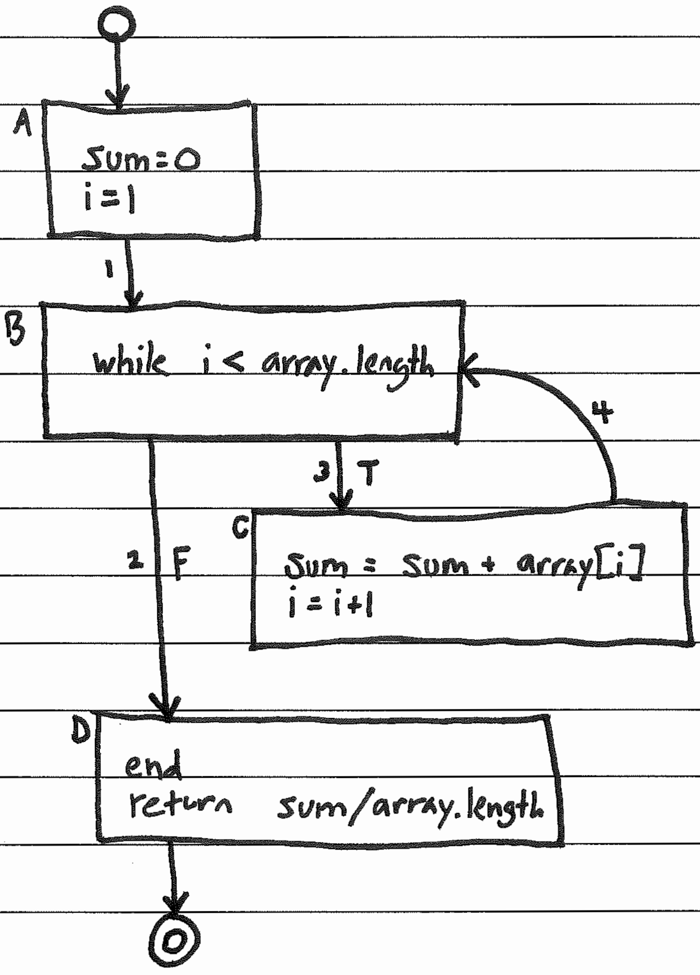
__U__ Should be fast (less than half a second)

10. [8pts] Using the fragments below, create a functional test (class and method) for the "index" page of a music-themed web app (which has the typical scaffold layout). The test should do the following in this order (1) retrieve user fixture "two" and sign in the user, (2) simulate an HTTP request for the index page, (3) check that the HTTP response does not report an error, (4) check that the rendered HTML table contains a cell with the famous singer of "Rolling in the Deep" and "Hello", and (5) check that the correct ERB is rendered (index.html.erb). Some fragments may be used more than once in your solution. Some fragments may not be used at all.

```
a) end
b) include Devise::Test::IntegrationHelpers
c) get albums_url
d) get album_url(@album)
e) assert_select "h1", "Album"
f) assert_select "td", "Adele"
g) assert_template :albums
h) assert_template :index
i) assert_response :error
j) assert_response :success
k) sign_in user
l) album = albums(:two)
m) user = users(:two)
n) test "should display albums" do
o) class AlbumsControllerTest < ActionDispatch::IntegrationTest
p) class Album < ApplicationRecord
```

o
b
n
m
k
c
j
f
h
a
a

11. [4pts] Draw a control-flow graph (CFG) for the function in Figure 1. In addition to the usual CFG features, label the nodes with capital letters (A, B, C, etc.), and label the edges with numbers (1, 2, 3, etc.). Don't forget to include entry and exit points.



Use the CFG you created for the function in Figure 1 to answer the following questions.

12. [2pts] Fill in the table below with a test suite that provides statement coverage. In the Covers column, list the letter labels (A, B, C, etc.) of the nodes covered by each test case.

Input	Expected Output	Covers
array		
[1,1]	1	A,B,C,D

13. [3pts] Fill in the table below with a test suite that provides branch coverage. In the Covers column, list the number labels (1, 2, 3, etc.) of the edges covered by each test case (only true/false edges needed).

Input	Expected Output	Covers
array		
[1,1]	1	3,2

14. [4pts] Fill in the table below with a test suite that provides path coverage. In the Covers column, list the number labels (1, 2, 3, etc.) of the edges covered by each test case. You need only cover executions that involve at most 1 iteration of each loop (if there are any). Before you fill in the table, list all the paths to be covered.

Paths:

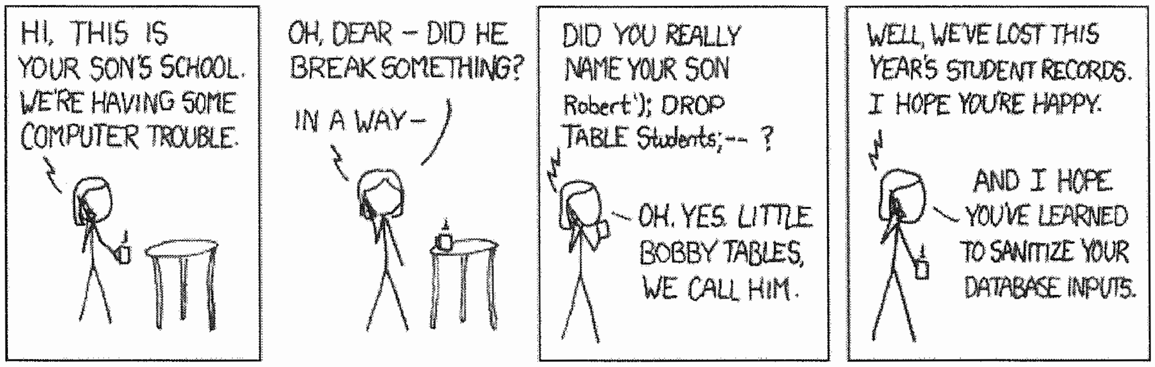
- 1,2
- 1,3,4,2

Input	Expected Output	Covers
array		
[1]	1	1,2
[1, 1]	1	1,3,4,2

15. [2pts] Which, if any, of your above three test suites would have caught the bug in the function?

All of the above test suites would have caught the bug.

16. [2pts] What type of attack did the parents in this XKCD comic perform?


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

a) Reverse lookup

b) Mask and shift

c) Cross-site scripting

[✓ d)] SQL injection

e) Child endangerment

17. [2pts] How do you prevent the above attack?

a) Interrupt requests

b) Merge tables

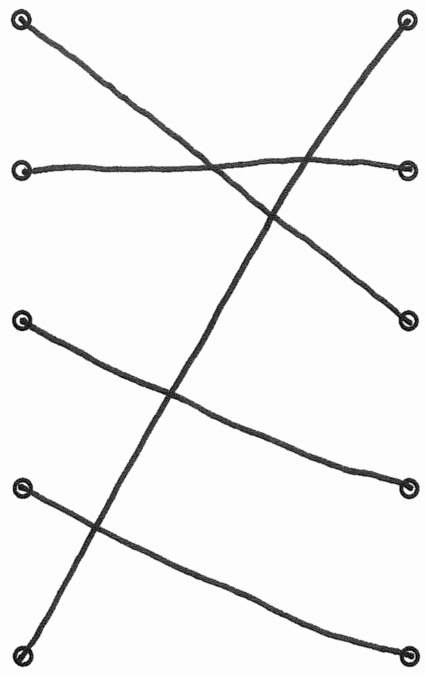
[✓ c)] Escape queries

d) All of the above

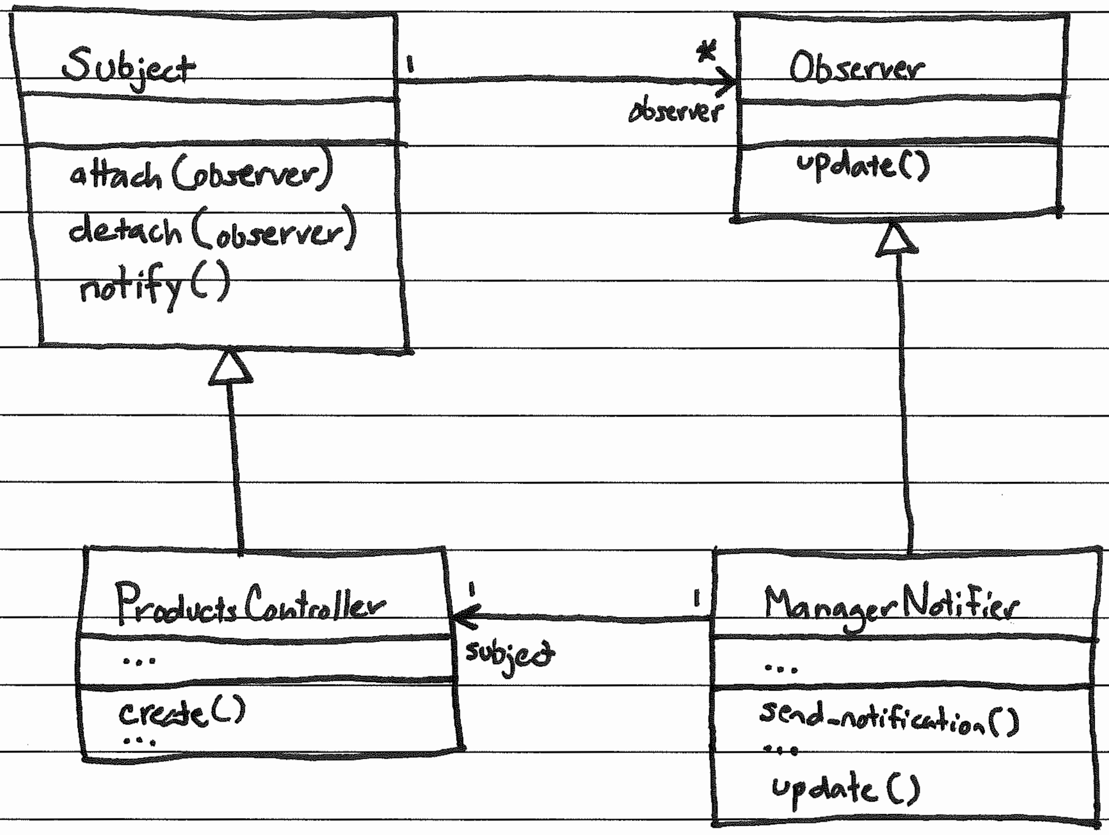
e) None of the above

18. [5pts] Match the Design Pattern to an example usage of the pattern.

Observer	You want your editor application to allow users to undo edits.
Singleton	Your point-of-sales application must log data from many different places in its code, but only one Logger object should do the logging for the entire system.
Adapter	Your *Doom* program needs to listen for keyboard and mouse clicks to manipulate how a player character moves, shoots, etc.
Builder	Your program must support switching among several different foreign language translation libraries (e.g., FrenchToX, HindiTranslatePro, English4Me), but each one has a slightly different interface.
Memento	Your application needs to generate large, complex PDF files (from scratch).



19. [8pts] Recall the Observer Design Pattern depicted in Figure 2. Imagine that you are designing a web app for product-supply business. Figure 3 depicts the classes that you have so far. In particular, you have designed a ProductsController class that records product information. As part of this controller's responsibilities, it must create new product entries when new products are added. You have also designed a ManagerNotifier that is capable of sending notification messages to managers at the company. The design problem you need to solve is how to make a ManagerNotifier "listen" for when a ProductsController creates a new product, and to send a notification to a manager whenever that happens. Draw a class diagram that applies the Observer Design Pattern to solve this problem. Use the same names used in the design pattern as much as possible (except make Ruby style). You must include all the classes from Figure 3 in your diagram (i.e., your changes should be additive). In particular, I expect that you will be adding classes, operations, inheritance relationships, and associations.



Figures

```
def average(array)
  sum = 0
  i = 1
  while i < array.length
    sum = sum + array[i]
    i = i + 1
  end
  return sum/array.length
end
```

Figure 1. Buggy function that computes the average value of an array of numbers.

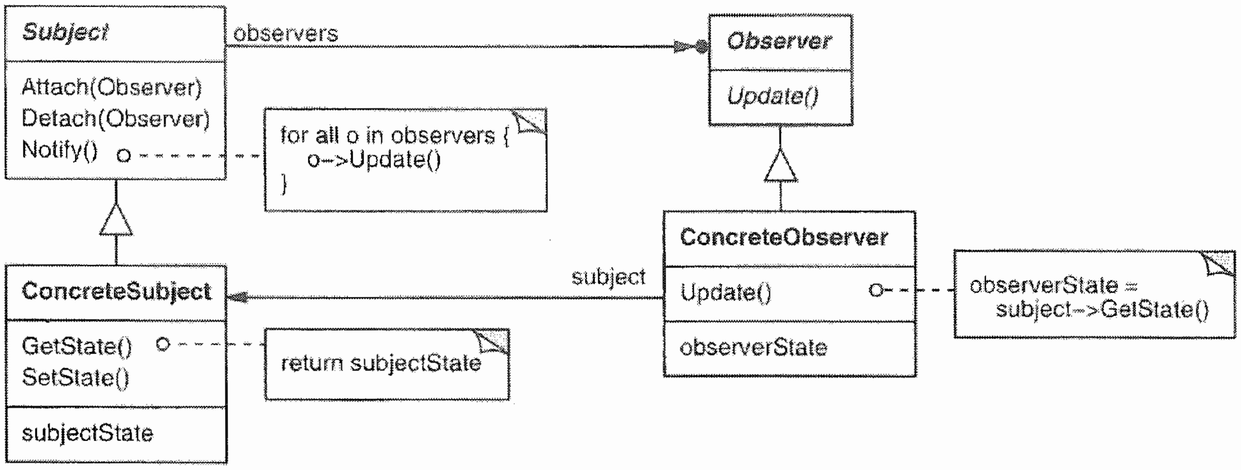


Figure 2. Observer Pattern from the "Gang of Four" book. (Note that the book uses an outdated class diagram notation.)

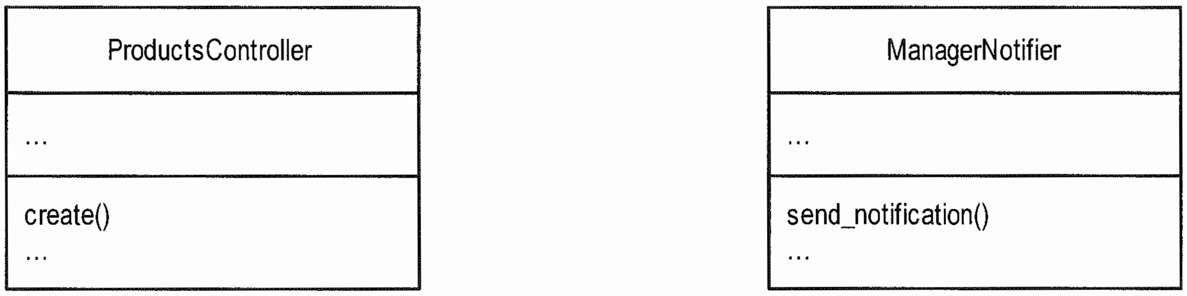


Figure 3. Classes for product-supply system.