Student Name____________________________Student Number____________

Concordia University
Department of Computer Science and Software Engineering

SOEN 6461 — Section SS Fall 2017 — Quiz 2 — Some Sample Questions

Special Instructions and Information

- This is a closed-book examination. You may not use any materials other than normal writing implements.
- Pocket calculators are not allowed.
- Turn off all electronic devices including cell phones.
- Write your name and student number at the top of this page.
- Place your student ID card on the desk face up.
- Answer all three questions. The marks for each question are indicated. The total marks are 60.
- Answer questions in the space provided on the examination paper. Write legibly. Write complete sentences. Answer the question that is asked.

Question 1

For each of the following statements, indicate whether the statement is true or false by circling T or F respectively.

(1) The GRASP Controller pattern determines which object receives and coordinates a system operation. **T or F**

(2) The GRASP Creator pattern always uses a Factory or Singleton pattern to create objects. **T or F**

(3) The Controller pattern sometimes suggests assigning control to a facade object. **T or F**

(4) The Controller pattern sometimes suggests assigning control to a session controller. **T or F**

(5) The Controller pattern sometimes suggests assigning control to a use case controller. **T or F**

(6) The owner of an object is a good candidate to be the creator of the object according to the Creator pattern. **T or F**

(7) The Information Expert for creation is a good candidate to be the creator of the object. **T or F**

(8) A layered architecture decomposes a software system into design patterns. **T or F**

(9) The GRASP Creator pattern leads to Low Coupling. **T or F**

(10) The GRASP Creator pattern leads to High Cohesion. **T or F**

Question 2 [10 marks]

Complete each sentence below by supplying the missing word(s) in the space provided. Use the word that best fits the meaning of the sentence.

1. GRASP stands for General Responsibility ________________ Software Patterns.
2. The Information Expert pattern says that the responsibility for a task should go to the class that has the ____________________ required to perform the task.
3. A module is a component of a system that provides services to other modules through an ____________.
4. The Proxy pattern provides a placeholder object for the ____________ object.
5. The ____________ object in the Blackboard architecture acts as a repository for ____________ and complete solutions to subproblems.
6. The R in RDD and the R in CRC stand for ____________.
7. Cohesion is a measure of how strongly ____________ the responsibilities of a class are, while Coupling is a measure of how strongly a class ____________ upon other classes.
8. The Strategy pattern is used when you wish to vary the ____________.

Question 3

(a) [5 marks] What is the Facade design pattern? What are the benefits of using a Facade for a subsystem or a layer?

Half-page answer required.

(b) [5 marks] What is a layered software architecture? What are the key features of the organization of the layers and how they communicate with one another? What is the responsibility of the i-th layer?

Half-page answer required.

Question 4

Brian Foote and Joseph Yoder in *"Big Ball of Mud"*, Fourth Conference on Pattern Languages of Programs, 1997 write

> *A BIG BALL OF MUD is haphazardly structured, sprawling, sloppy, duct-tape and bailing wire, spaghetti code jungle. We've all seen them. These systems show unmistakable signs of unregulated growth, and repeated, expedient repair. Information is shared promiscuously among distant elements of the system, often to the point where nearly all the important information becomes global or duplicated. The overall structure of the system may never have been well defined. If it was, it may have eroded beyond recognition.*
>
> *What does this muddy code look like to the programmers in the trenches who must confront it? Data structures may be haphazardly constructed, or even next to non-existent. Everything talks to everything else. Every shred of important state data may be global. Where state information is compartmentalized, it may be passed promiscuously about though Byzantine back channels that circumvent the system's original structure.*
>
> *Variable and function names might be uninformative, or even misleading. Functions themselves may make extensive use of global variables, as well as long lists of poorly defined parameters. The function themselves are lengthy and convoluted, and perform several unrelated tasks. Code is duplicated. The flow of control is hard to understand, and difficult to follow. The programmer's intent is next to impossible to discern. The code is simply unreadable, and borders on indecipherable. The code exhibits the unmistakable signs of patch after patch at the hands of multiple maintainers, each of whom barely understood the consequences of what he or she was doing. Did we mention documentation? What documentation?*

(a) [5 marks] For each example of poor use of cohesion in the text of in Foote and Yoder, discuss why it is poor cohesion and how you could improve it in the design.

Half-page answer required.