



Carleton
UNIVERSITY

FINAL

EXAMINATION
SUMMER 2012

DURATION: 3 Hours

No. of Students: 20

Department Name & Course Number: Systems and Computer Engineering
SYSC-3020

Course Instructor (s) : Dr. Greg Franks

AUTHORIZED MEMORANDA $8\frac{1}{2} \times 11$ **sheet, double sided**

Students **MUST** count the number of pages in this examination question paper before beginning to write, and report any discrepancy to a proctor. This question paper has 9 pages + cover page = 10 pages in all.

This examination question **may not** be taken from the examination room.

In addition to this question paper, students require: an examination booklet **no**
Scantron Sheet **no**

Name: _____

Student Number: _____

Instructions:

1. This exam has a total of 115 marks and contributes to 50% of your final course mark.
2. Answer all questions.
3. **Please be brief** when answering questions. You might run out of time if you are too wordy. Single words are quite acceptable for use case and class names.
4. **You are doing the design.** Don't ask the proctor for clarification. If necessary, make assumptions, but be sure to write them down along with your answer.
5. Use the number of marks allotted for each question to gauge the amount of time you should devote to that question.

Question 1

15 Marks

Generalization is the most tightly coupled form of relationship between two elements.

1. **Five** different types of *generalization* were discussed in class. What were they and *briefly* describe their characteristics paying close attention to whether the generalization approach is either good or bad. [10 marks]

i)

ii)

iii)

iv)

v)

2. In one paragraph, describe the *Open-Closed Principle*.

[5 marks]

Question 2

34 Marks

You are to design a class diagram for the Department of Systems and Computer Engineering Fourth Year Project database. The system is to store proposals, projects, students and professors.

A **proposal** lists the *title* and a *description* of the work for a project. It can be supervised by one or two professors. Each proposal is unique.

A **project** is always based on a proposal. One or more projects can be based on the same proposal. Students join projects. A project needs a least one student, but is limited to no more than four students in total.

A **student** has a *name*, an *email address*, and a *student number*. Students can join projects.

A **professor** has a *name* and an *email address*. Professors create proposals.

- a) Draw a class diagram for the Fourth Year Project System described here including multiplicities, roles, attributes, and operations. Be sure to denote the scope for attributes and operations, and to show which operations and attributes are class (or factory) and those which are object (or instance).

[14 marks]

- b) Draw a sequence diagram for the following use case for registering for a project, including the necessary boundary and control classes. [20 marks]

Title: Register in Project

Actors: Student

Pre Condition: Student has authenticated.

Steps

- 1) Find all proposals.
- 2) Display the list of available proposals.
- 3) The student chooses one of the available proposals, (p).
- 4) Find all projects currently associated with this proposal.
- 5) Find the first project which is not full (q).
- 6) Register the student with the project q .

Alternate Courses:

- a) At step 5, there may not be any available projects associated with the proposal for the student to join. In this case, create a new project and associate the student with this project.
- b) At step 5, all of the projects associated with the proposal are full. In this case, create a new project and associate the student with this project.

Exit Condition: The student has registered for a project.

Question 3

20 Marks

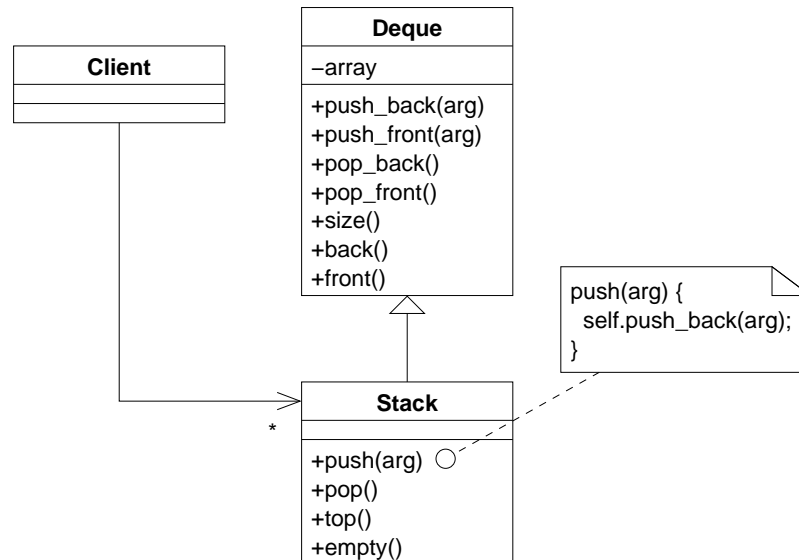
You are to design a state machine for some traffic lights at the intersection of Bronson Avenue and Sunnyside Street. The output of the state machine is a pair of colour codes, i.e., $\{red:green\}$ to indicate that the traffic light on Bronson is *red* and the traffic light on Sunnyside is *green*. The normal cycle of the lights is to show *green* on Bronson for 90 seconds, then show *yellow* on Bronson for 10 seconds, then show *red* on both Bronson and Sunnyside for 2 seconds. The control then shows *green* on Sunnyside for 30 seconds, followed by *yellow* for 10 seconds, and then *red* on both Sunnyside and Bronson for 2 seconds. While one street is showing either *green* or *yellow*, the other street will be showing *red*.

The traffic light also has a button to allow pedestrians to cross Bronson. If the pedestrian pushes the button while the light is green for traffic traveling on Bronson Ave, and the light has been green for at least 30 seconds, the normal 90 second interval will be preempted and the lights will change. If the light has not been green for 30 seconds, pushing the button will cause the light to change *after* 30 seconds have elapsed. In other words, traffic on Bronson will see a green light for at least 30 seconds, but no more than 90. Pushing this button anywhere else in the cycle of the state machine has no effect.

Question 4

8 Marks

The figure below shows the design of class **Stack** supporting the public operations `push()`, `pop()`, `top()` and `empty()`. This **Stack** class is implemented by subclassing **Deque** (a double ended queue). **Deque**'s have the operations `push_back()`, `pop_back()`, `push_front()` and `pop_front()`.



1. From the standpoint of software engineering the approach shown using specialization here is considered a poor design. Why? [2 marks]
2. Draw a class diagram showing the design of a stack which **does not specialize a deque**; rather, it uses delegation. You only need to show the operations `push()` and `push_back()`; you do not need to include the operations shown in the diagram here. Don't forget to include any other labels for associations or attributes that you feel you need. [4 marks]

3. Write the function `push()` using the labels you used in your class diagram. Use a syntax similar to the comment in the class diagram here (i.e., C/Java-like). [2 marks]

Question 5

6 Marks

For each design goal below, indicate whether the MVC architecture helps or hurts. Briefly justify each answer.

i) extensibility of the system:

ii) response time:

iii) modifiability of the design:

Question 6

4 marks

In a couple of sentences, provide a comparison of the Adapter and Facade design patterns. (You can use diagrams to support your answer.)

Question 7

4 marks

The Decorator and Composite design patterns are structurally (class diagram) very, very similar (almost identical). In a couple of sentences, provide a comparison of the Decorator and the Composite design patterns. (You can use diagrams to support your answer).

Question 8

6 marks

Using the example of the Trip Planning Software discussed in class, suppose that the **PlanningService** class needs to sort trips using either one of the following algorithms: a bubble sort, a heap sort, or a shell sort; and the use of an algorithm will depend on the context of the use of the collection of trips: sometimes a bubble sort should be used, sometimes a heap sort should be used.

1. Which design pattern would you use to facilitate this functionality, i.e., allowing the use of any of those sorting algorithm, while allowing the definition of new such algorithms in future releases?
[2 marks]

2. Once you have selected a design pattern, and justified its use (by justifying that the needs correspond to the intent of the pattern), show in a class diagram what the use of the pattern would look like. [4 marks]

Question 9

18 marks

1. What is Software Verification and what is Software Validation? [2 marks]

2. What is a stub? [2 marks]

3. What is a driver? [2 marks]

4. What is an oracle? [2 marks]

5. What is referred to as a Black-box testing technique, what is referred to as a White-box testing technique? [2 marks]

6. What are the advantages of Black-box testing? [4 marks]

7. What are the advantages of White-box testing? [4 marks]