Name Student ID Signature

Department of Electrical and Computer Engineering, McGill University

ECSE 321 - Introduction to Software Engineering, Fall 2006

Final Examination

December 20, 2006, 14:00-17:00

Examination version III

Examiner: Prof. Shie Mannor

Associate Examiner: Prof. Warren Gross

INSTRUCTIONS

1. Write your name, McGill student ID number, and signature in the top of this page. If you require an answer booklet, write your name there too.
2. This examination consists of two sections:
    a. Section I consists of 13 multiple-choice questions.
    b. Section II consists of 6 open questions.
3. Section I is worth 39 marks in total; section II is worth 61 marks in total. The multiple-choice questions are equally weighted; wrong or omitted answers are worth 0 marks.
4. This examination booklet has 16 pages, including this page. Keep all pages together.
5. Please use only the space allocated in the booklet.
6. Multiple choice questions: circle the correct option in multiple choice questions. Any ambiguity in the final answer will be considered invalid. If you change your mind, you may write "Final: A" for making sure we understand you think that A is the correct answer. Some questions require extra explanation. Make sure you explain your answer. In these questions the explanation is worth more than guessing the correct answer.
7. Several questions require a written explanation. Be concise and precise. Don't use more space than allocated to you.
8. You may use any written material you wish (books, notes, lecture notes, tutorial notes, etc.). You may not use a computer in this exam.

Good luck!

Section I: Multiple choice questions

(3 marks for each question, total of 39 marks)

1. In the final product of RMS^2 multiple users should be able to access the system at the same time. Suppose that if it happens that a user books a room while an administrator edits the room properties or an operator books maintenance then the user will be notified after booking that his booking was not successful (the user is made aware of this possibility as part of the usage agreement). This situation is a:

A. Deadlock
B. Race
C. Livelock.
D. Starvation
E. None of the above.

Explain: (use allocated space only) ____________________

____

____

2. Suppose we decide to use the exponential growth model for bug detection. After 1 week of debugging we discover 100 bugs. After another week we find another 50 bugs. The approximate number of bugs we started with is: (For your convenience, ln (2) ~ 0.7.)

A. 150.
B. 300.
C. 450.
D. 500.
E. 600.

Explain: (use allocated space only) ____________________

____

____

3. Which of the following is true?

A. The proxy design pattern facilitates easier testing because we do not have to instantiate complex objects.
B. The composite design pattern cannot be used if we don't know the depth and width of the object hierarchy in advance.
C. Every subsystem must have the Façade pattern.
D. The strategy pattern and the bridge pattern have similar UML diagrams, but serve a different purpose.
E. The adapter pattern is used heavily for testing.

4. Suppose a program takes several command line arguments which are used as parameters throughout the code. What is the **best** way to store and access the values of these arguments?

A. Pass the values as constructor arguments to classes that require these values.
B. Create a class which static variables and methods so that the values can be accessed from anywhere.
C. Create a class using the Singleton design pattern which will hold all global variables.
D. Create a class using the Abstract Factory design pattern which will hold all global variables.
E. Store the information in a XML file and use the Bridge design pattern to access the information.

5. When a system is said to have 4 9's reliability it means that it can afford to be not working for (for your conveniences: there are 86400 seconds in a day, 525600 minutes in a year, and 43200 minutes in a month):

A. About 50 seconds a day.
B. About 10 minutes a year.
C. About 10 hours a year.
D. About 10 minutes a day
E. About 5 minutes a month

6. The Arena Case Study: Suppose we decided to limit the architecture of the Arena system to turn-based games (i.e., games where each player moves at his/her turn). What effect would that have on the system architecture?

A. We could use a peer-to-peer architecture for the whole system.
B. We would abolish the need for the MatchFrontEndPeer adapter.
C. The ArenaClient subsystem would become much more complex.
D. We could use a client-server architecture.
E. The three-tier architecture of the game organization part would become obsolete.

Explain: (use allocated space only)

7. The company you are working for is creating an ATM. You are working on the user interface where the user will enter his/her personal identification number (PIN). Which of the following requirements is a valid functional requirement?

A. With the use of the keypad, the user shall enter a 4 to 6 digit pin.
B. The system will verify the validity of the user by comparing the entered PIN with the Oracle database for the given user.
C. If the user is unable to correctly login to the system after 3 tries, no more attempts will be allowed and the card will be swallowed. (Swallowed will mean that the card is not returned, and destroyed.)
D. Once the user has entered his/her pin, the system shall verify the validity within 1000ms 90% of the time.
E. The login screen will be appealing.

8. Bob is designing a role-playing game, where a player can be a fighter, mage, priest, or druid. Bob has decided to make each as a subclass of player, but can't really figure out what to do with the druid. A druid can shapeshift (change form whenever he/she wishes) and be a bear, wolf, rat, or human. Each of them will have the same functions, but perform them differently. What do you suggest he does?

A. Use a Factory pattern.
B. Use a Template Method pattern.
C. Use a State pattern.
D. Use a Singleton pattern.
E. Use the composite pattern.

9. Which of the following is a benefit of the pipe-and-filter architecture?

A. It reduces cohesion.
B. It provides abstraction.
C. It increases reusability.
D. It improves runtime efficiency.
E. None of the above.

10. Which of the following is **not** important with software usability?

A. Learnability.
B. Efficiency.
C. Memorability.
D. Error recoverability.
E. None of the above.

11. After completing the RMS^2 system successfully the client is happy with the user interface, and decides that auctioning rooms on the Internet is the best source of future revenues. You are hired to build an auctioning systems based on the RMS^2 system. The interface should be web-based. User registration will be done over the web. The customer is mostly concerned about the reliability of the systems and its ability to sustain high load for the many complexes the customer owns worldwide. The customer is willing to work with you on the reliability and performance issues. What kind of software life-cycle will you use?

A. Waterfall model.
B. Spiral model with one iteration.
C. Saw-tooth model.
D. Extreme programming model.
E. V-model

Explain: (use allocated space only)

12. Which one of the following statements is FALSE?

A. If you create a set of tests that achieve 100% coverage of the code, it ensures that the software has been fully tested.
B. Unit testing can be both black box and white box testing.
C. Development of the test plan can start before the system design is available.
D. A significant portion of the bugs are detected during alpha and beta testing.
E. White box testing and black box testing can find the same bug.

13. When is it a better idea to use delegation instead of inheritance?

A. When two objects are involved in handling a single request.
B. When we don't want the receiving object to misuse the delegate object.
C. Never, because delegation can be done with a single class, while a class can inherit from multiple classes.
D. Almost always because delegation is more straightforward.
E. Almost always because delegation is more efficient.

Section II

Question 1 – Analysis (state diagrams)

(8 marks)

Automatic Coffee Machine

Consider the following description of an automatic coffee machine:

An automatic coffee machine can deliver different types of coffees (espresso, latte, and cappuccino) in exchange for a given amount of money (different drinks may cost differently). The user may add sugar, milk, and cream (any combination of these is possible). You can assume that the user can press a button for every drink and that there is a small screen that presents prices. The user first chooses the drinks, then enter the money (exact amount is not needed) after observing the price. After the drink is poured, the user can add sugar/crème/milk.

Draw a state diagram for the coffee machine. You may ignore problematic/error situations in your diagram.

Question 2 – Coding Conventions

(6 marks)

In the following function, which lines do not follow proper coding conventions?

```
01 public static long power (long x, long y)
02 {
03     long t;
04
05     if (y == 0 || x == 1)
06         t = 1;
07     else if (y == 1) {
08         t = x; }
09     else {
10         t = power (x, y / 2);
11         t *= t;
12         if (y % 2 == 1) t *= x;
13     }
14     return t;
15 }
```

List the coding conventions error below.

Line number	Error explanation
01	x and y have meaningless names – should be base and exponent

Question 3: Testing

(8 mark)

Bob is designing a role-playing game, and one of its features is that every character can select his/her own name. Names must be at least 2 characters long, but not more than 12. They must start with a capital letter and end with an alphanumeric character. Anything typeable on a keyboard is allowed for the middle characters. Out of the following test cases, which are success cases, boundary cases, and failure cases. Also state all sets of equivalence classes, if any.

a. "Jonny"
b. ""
c. "J"
d. "Jo"
e. "BabeRuth"
f. "Shie "
g. "Jon & Fred"
h. "Jon1"
i. "jon"
j. "Jon & Freddy"
k. "Jojol Freddy"
l. "J...o"
o. “Jingaloo walla”

Classify the test cases below:

Success: a, ____

Failures: b, ____

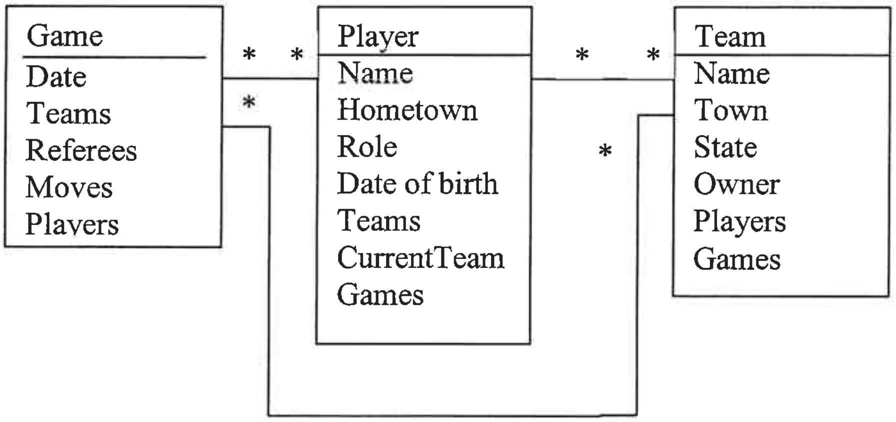
Boundary: b, ____

Equivalence classes: ____

Question 4: Data Schema

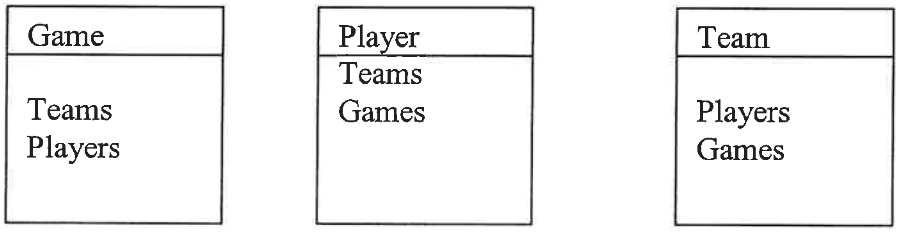
(12 marks)

Consider a data schema for the following class diagram:



1. Describe the data schema tables for the above diagram as two dimensional tables (relational database style). You do not need to specify a table for Town or Owner. Use the space below (write the name of every table and identify its columns).

2. Suppose that we want to eliminate the many-to-many associations and decide to add repository classes. Complete the class diagram below (you do not have to draw a class for Owner or Town, add classes and associations to the diagram below just to show how the associations would work)

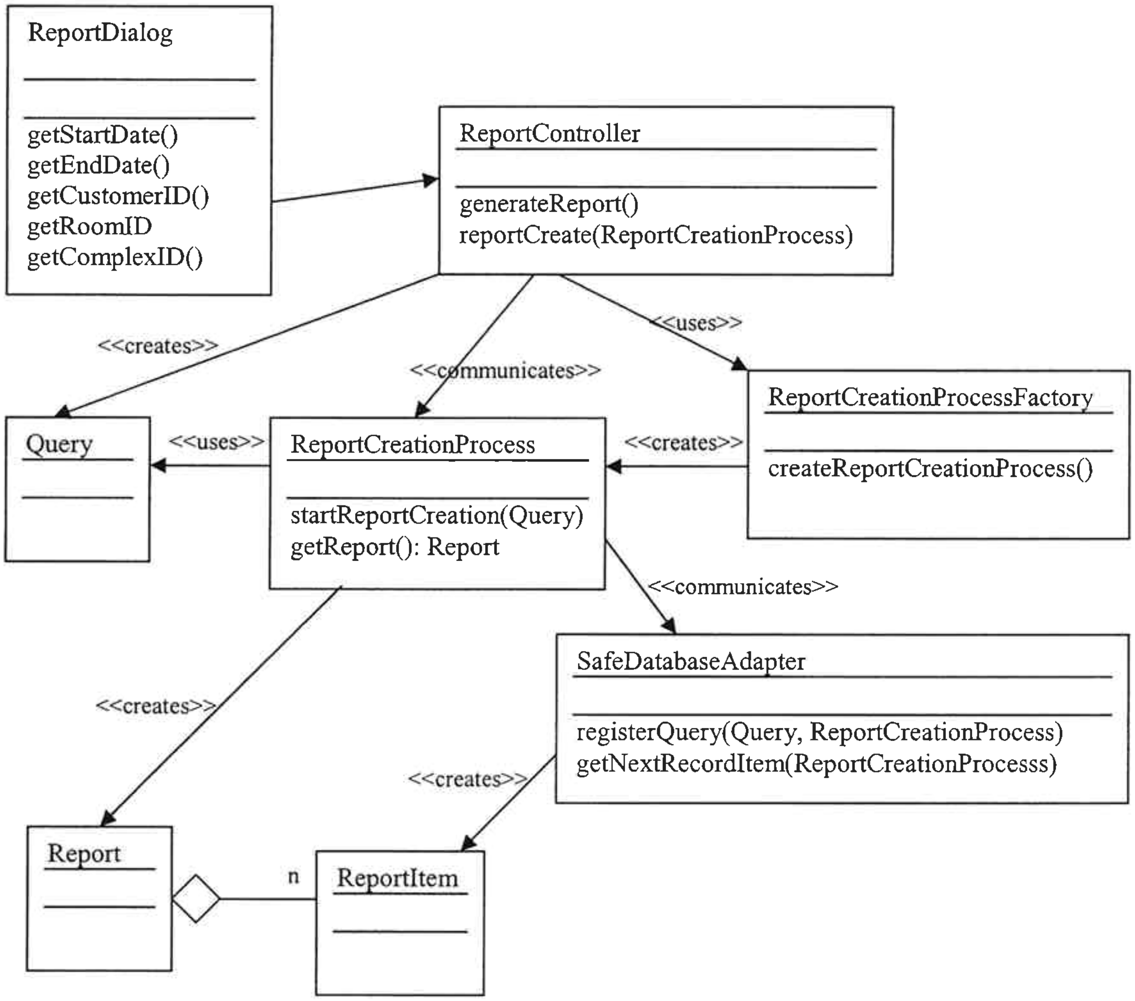


3. Describe the new data schema tables.

Question 5: Use cases, Sequence Diagrams, Integration

(15 marks)

In this question we consider generating reports for the final RMS^2 system. This system is supposed to support multiple users running reports simultaneously. The process of generating reports may take a few minutes so that efficiency is a prime concern here. Below is the proposed class diagram.



The classes are:

- *Report*: a report
- *ReportItem*: a single line in the report
- *ReportDialog*: A user interface window that is used to order a report
- *ReportController*: A control class that is used for generating a report. The method that is called after the request of the user to create a report is generateReport()

- *Query*: a class that contains the parameters according to which the report is created.
- *SafeDataBaseAdapter*: This class is an interface between the database and objects of the type *ReportCreationProcess*. This interface is safe in the sense that multiple objects of the type *ReportCreationProcess* can communicate with it simultaneously. This class is a singleton and it contains two methods:
  - registerQuery(Query q, ReportCreationProcess p): this method registers the object p as the source of the query whose parameters are q. Calling this method is essential before calling the method getNextReportItem(ReportProcess p).
  - getNextReportItem(ReportProcess p): this method generates the next line (ReportItem) in the report for a Query and ReportCreationProcess that were created using the RegisterQuery method. This method returns null if there are no additional lines for the query.
- *ReportCreationProcess*: An object that communicates with the *SafeDataBaseAdapter* class and is responsible to creating the report. For this end, a method called startReportCreationQuery(Query q) is defined. This method has a loop which repeatedly calls the method getNextReportItem() of the class *SafeDataBaseAdapater*.
- *ReportCreationProcessFactory*: an object that is responsible for creating objects of the type *ReportCreationProcess*, and is the only way to create such objects. The method createReportCreationProcess is defined.

Notes:

1. Additional private methods may exist.
2. There are no other public methods except for constructors and the methods that appear in the diagram.
3. Additional attributes may exist.
4. There are no additional associations beyond those that appear in the diagram.
5. The functionality of methods that appear in the diagram is to be deducted based on their names.
6. Methods may have additional parameters beyond the parameters described above.

A. Draw the sequence diagram that is used to produce a report (pick a typical sequence without exceptions).

B. Suggest an integration plan of the above classes (which classes will be integrated first and in what style). Justify your suggestion. You may draw a diagram below, you feel this is helpful for explaining the integration method.

Question 6: Design patterns

(12 marks)

Consider the following Java program.

```
import java.util.*;

interface DrawingAPI {
    public void drawCircle(double x, double y, double radius);
}

class DrawingAPITake1 implements DrawingAPI {
   public void drawCircle(double x, double y, double radius)
   { System.out.printf("APITake1.circle at %f:%f radius %f\n", x, y,
radius); }
}

class DrawingAPITake2 implements DrawingAPI {
   public void drawCircle(double x, double y, double radius)
  { System.out.printf("APITake2.circle at %f:%f radius %f\n", x, y,
radius); }
}

interface Shape {
   public void draw();                                 // low-level
   public void resizeByPercentage(double pct);         // high-level
}

class CircleShape implements Shape {
   private double x, y, radius;
   private DrawingAPI drawingAPI;
   public CircleShape(double x, double y, double radius, DrawingAPI
drawingAPI) {
       this.x = x;  this.y = y;  this.radius = radius;
       this.drawingAPI = drawingAPI;
   }

   public void draw() { drawingAPI.drawCircle(x, y, radius); }
   public void resizeByPercentage(double pct) { radius *= pct; }
}

class DrawingClient {
   public static void main(String[] args) {
       Shape[] shapes = new Shape[2];
       shapes[0] = new CircleShape(1, 2, 3, new DrawingAPITake1());
       shapes[1] = new CircleShape(5, 7, 11, new DrawingAPITake2());

       for (Shape shape : shapes) {
           shape.resizeByPercentage(2.5);
           shape.draw();
       }
   }
}
```

A. Draw a class diagram for this code. (Use allocated space only.)

B. What design pattern is used in the Java program?

C. What will the output of the program be?