

CSC301 - Introduction to Software Engineering
Summer 2008 - Course Exam
University of Toronto (St. George)
Monday August 11th, 2008

Duration: 3 hours

Open Book and Notes

Last Name: _____ First Name: _____

Student No. : _____ (Please write name & student no. on every page)

Do **not** turn this page until you have received the signal to start. In the meantime, please fill out the identification section above, and read the instructions below carefully.

This exam consists of 6 questions (numbered 0 to 5) on 13 pages (*including this one*), printed on one side of the paper. When you receive the signal to start, please make sure that your copy of the examination is complete.

Answer each question directly on the examination paper, in the space provided, and **use the reverse side of the page for rough work**. If you need more space for one of your solutions, use the reverse side of the page and indicate **clearly** the part of your work that should be marked.

In the programming questions, you do not have to write pre-conditions/post-conditions. Also, you do not have to comment your code but use **meaningful** variable names. You are allowed to use helper procedures.

Be aware that concise, well thought-out answers will be rewarded over long rambling ones. Also, unreadable answers will be given zero (0) so please write legibly.

- 0. _____ / 3 (Your Name & ID on every page)
- 1. _____ / 26 (Short Questions)
- 2. _____ / 16 (Software Engineering Rules and Guidelines)
- 3. _____ / 12 (Refactoring)
- 4. _____ / 33 (Object Oriented Paradigm)
- 5. _____ / 10 (Design Patterns)

_____ / 100 **TOTAL**

Question 0. [3 marks]

Write your first initial and last name, and your student no. at the top of every page.

Question 1. Short Questions [26 marks total]

(a) [1 mark] What word is used to describe software that can easily have new features and capabilities added onto it without requiring major modification of the source code?

- i. Fantastic
- ii. Extensible
- iii. Integratable
- iv. Open

(b) [1 mark] What type of UML diagram is used to describe a specific scenario of a use case?

(c) [2 marks] The software development methodology known as *The Waterfall Model* is an iterative approach that breaks a problem down into a number of smaller individual development cycles. Is this true/false? Why?

(d) [2 marks] *Extreme Programming* says that you should start with designs that are as detailed as possible, so that they can be reused for different programs. Is this true/false? Why?

(e) [2 marks] A primary purpose of the Visitor pattern is to (*choose one answer only*):

- ☐ expose the representation of a data structure so it can be traversed more easily
- ☐ ensure that the post-condition of a while loop implies the loop's test expression
- ☐ hide the code required to traverse some representation of a data structure
- ☐ retrofit message-passing buzzword technology onto nominally late-binding APIs
- ☐ make it impossible for client code to traverse complex data structures

(f) [2 marks] What is the normal order of execution of test plans? (*choose one answer only*)

- ☐ Acceptance, Integration, System, Unit
- ☐ Acceptance, System, Integration, Unit
- ☐ System, Acceptance, Integration, Unit
- ☐ Unit, System, Integration, Acceptance
- ☐ None of the above

(g)[2 marks] Software Engineering theories are (*choose one answer only*):

- ☐ Folklore
- ☐ Based on scientific experimental data
- ☐ Mostly extrapolations from project experience and some based on scientific data
- ☐ Processes that focus on object-oriented systems

(h)[2 marks] The prototyping model of software development is (*choose one answer only*):

- ☐ A reasonable approach when requirements are well defined
- ☐ A useful approach when a customer cannot define requirements clearly
- ☐ The best approach to use for projects with large development teams
- ☐ A risky model that rarely produces a meaningful product

(i) [2 marks] What role do user stories play in agile methodology? (*choose one answer only*)

- ___ Define useful software features and functions delivered to end-users
- ___ Determine a schedule used to deliver each software increment
- ___ Provide a substitute to performing detailed scheduling of activities

(j) [2 marks] Consider the following list of cohesion types:

- i. sequential
- ii. procedural
- iii. utility
- iv. temporal

List the four types of cohesion given above in order from LOWEST cohesion to HIGHEST cohesion.

(k) [2 marks] Is it possible that a program passes a functional black-box test, but still is not of good quality? Why?

(l) [6 marks] Order the following software development processes:

Waterfall model, Agile development model, Iterative development model

in terms of:

- i. Time to possibility for customer feedback (fastest → slowest)
- ii. Quality of the design/implementation of the first release (best → worst)
- iii. Number of features included in first release (largest → smallest)

Question 2. Software Engineering Rules and Guidelines [16 marks total]

(a) [10 marks] Following is the script of a meeting in a software company. The attendants are Tom (Senior Team Leader), Chen (Senior QA), Lisa, Waleed, and David (Team Leaders). During this discussion, which software engineering rules/laws were ignored? Write the name of the person and the relevant rule/law which contradicts the suggested action.

Tom: hello everyone, as you know, we are facing a very critical situation, 4 of the 36 requirements were mistakenly dropped from the project in the design phase. The management is still investigating how did this happen. The client is upset and is threatening to sue. I wanted your feedback regarding the feasibility of implementing those missing features and including them in the current release.

Waleed: the 5 sub-teams are all overloaded with work. Why don't we hire 4 new contractors to help us in coding the missing features?

Tom: ya, that's definitely one possibility.

Lisa: actually, I heard of a new tool that reads a UML design and generate the code automatically. Instead of getting extra developers, why don't know we use the money to buy the tool and that should speed the development.

David: I think we don't need any consultants, we can communicate to the sub-teams how critical this situation is and ask everyone to put more hours.

Tom: yes, we can do that as well. Actually, I am going to pursue the 3 routes, Lisa please send me the URL to the software you mentioned.

Chen: there is one more issue I wanted to discuss.

Tom: go ahead.

Chen: we have been getting an increasing number of false bug reports from the client. I think once they realized that there are missing features, they started to over-react. The software is very stable and I would like to move two of the dedicated QA to another project.

Tom: ok, I am fine with that, but let us have a short meeting after this to discuss the risks.

(b) [4 marks] What type of coupling is in the following code snippet? How can it be improved?

```
public void draw(String key)
{
    if( key.equals("drawCircle") )
    {
        drawCircle();
    }
    else
        if( key.equals("drawRectangle") )
        {
            drawRectangle();
        }
        //else...
}
```

(c) [2 marks] Rewrite the following Java code using Hungarian notation (*you may simply fix the variable names instead*).

```
public class Painter extends Canvas
{
    protected Color    background,
                    foreground;
    protected Vector    Coordinates;
    protected Hashtable Actions;

    public void paint( Graphics graphics )
    {
        int Size;
        graphics.setColor( background );
        Size = Coordinates.size( );
        for( int Index; Index < Size; Index++)
            graphics.drawRectangle((Dimension)Coordinates.elementAt(Index));
    }
}
```

Question 3. Refactoring [12 marks total]

Each of the following code snippets requires refactoring. Rewrite the code and indicate which refactoring rule you used.

(a) [3 marks]

```
int getRating()
{
    if( moreThanFiveLateDeliveries( ) )
        return 2;

    return 1;
}

boolean moreThanFiveLateDeliveries()
{
    return _numberOfLateDeliveries > 5;
}
```

(b) [3 marks]

```
public class Employee {
    ....
}

public class Salesman extends Employee {
    ....
    public String getName( ){.....}
}

public class Engineer extends Employee {
    ....
    public String getName( ){.....}
}
```

(c) [3 marks]

```
Customer customer = site.getCustomer();
BillingPlan plan;
if (customer == null)
    plan = BillingPlan.basic();
else
    plan = customer.getPlan();

String customerName;
if (customer == null)
    customerName = "occupant";
else
    customerName = customer.getName();

int weeksDelinquent;
if (customer == null)
    weeksDelinquent = 0;
else
    weeksDelinquent = 10;
```

(d) [3 marks]

```
int discount(int inputVal, int quantity, int yearToDate)
{
    if(inputVal > 50)
        inputVal = inputVal - 2;

    .....
    .....
    .....
}
```

Question 4. Object Oriented Paradigm [33 marks total]

(a) [3 marks] One technique we have talked about to identify classes is to use nouns/verbs/adjectives in the problem statement. Briefly describe this technique.

(b) [15 marks]

The following paragraph is part of a requirements document for a resource management software to be used in a university;

The campus has a collection of classrooms. Each classroom has a collection of equipments, which may include microphone, blackboard, overhead projector, TV, Internet connection, etc. Each equipment has information such as history, condition, available forms (built in or for picking up somewhere), as well as information specific to the equipment, such as size for blackboard and TV, mobile or not for microphone, and speed of connection to Internet. Each classroom has a list of courses scheduled to take place in it, the instructor for each course, requested equipments and setups for each course, and a manager in charge of setting up the equipments. The system should allow instructors to look up the list of equipments in a classroom, request equipments and setups, and query the setup status. The system should send request for equipments and setups in a classroom to the classroom manager. The manager can update equipment information and setup status. In case of an equipment cannot be setup according to the request, the manager should notify the instructor as soon as possible. The system should also allow anyone to send comments to the manager.

Using the nouns/verbs/adjectives method, perform an object oriented analysis and construct a UML class diagram for the entities identified (*show classes, attributes, inheritance, composition and aggregation where applies*). Document your assumptions.

This page is intentionally left blank for your answer to question 4 (b)

The campus has a collection of classrooms. Each classroom has a collection of equipments, which may include microphone, blackboard, overhead projector, TV, Internet connection, etc. Each equipment has information such as history, condition, available forms (built in or for picking up somewhere), as well as information specific to the equipment, such as size for blackboard and TV, mobile or not for microphone, and speed of connection to Internet. Each classroom has a list of courses scheduled to take place in it, the instructor for each course, requested equipments and setups for each course, and a manager in charge of setting up the equipments. The system should allow instructors to look up the list of equipments in a classroom, request equipments and setups, and query the setup status. The system should send request for equipments and setups in a classroom to the classroom manager. The manager can update equipment information and setup status. In case of an equipment cannot be setup according to the request, the manager should notify the instructor as soon as possible. The system should also allow anyone to send comments to the manager.

(c) [15 marks]

Based on your answer to part (b), perform an object oriented design for the resource management software in which you use the following patterns: command, observer, iterator, and object pool. Construct a UML class diagram for the entities identified (*show classes, attributes, inheritance, composition and aggregation where applies*). Document your assumptions.

The campus has a collection of classrooms. Each classroom has a collection of equipments, which may include microphone, blackboard, overhead projector, TV, Internet connection, etc. Each equipment has information such as history, condition, available forms (built in or for picking up somewhere), as well as information specific to the equipment, such as size for blackboard and TV, mobile or not for microphone, and speed of connection to Internet. Each classroom has a list of courses scheduled to take place in it, the instructor for each course, requested equipments and setups for each course, and a manager in charge of setting up the equipments. The system should allow instructors to look up the list of equipments in a classroom, request equipments and setups, and query the setup status. The system should send request for equipments and setups in a classroom to the classroom manager. The manager can update equipment information and setup status. In case of an equipment cannot be setup according to the request, the manager should notify the instructor as soon as possible. The system should also allow anyone to send comments to the manager.

Question 5. Design Patterns [10 marks total]

(a) [4 marks] A developer tried to define the QuittingClient class below using the Singleton design pattern, but did not get it right!

i) [2 marks] What is wrong with the implementation? (*you may simply add corrections to the code if you like*).

ii) [2 marks] What is the result of this mistake? (*i.e., describe how this affects use of the QuittingClient class*).

```
import java.awt.*;
import java.awt.event.* ;

public class QuittingClient extends Frame
                                implements ActionListener {

    private QuittingClient    _client    = null;
    private Button            _exit      = null;
    private Panel              _panel     = null;

    private QuittingClient( ) {
        super("Exit Example With Singleton");

        _exit = new Button("Exit");
        _exit.addActionListener( this );

        _panel.setLayout( new BorderLayout( ) );
        _panel.add("Center", _exit );

        add( _panel );
    }

    public QuittingClient getClient( ) {
        if ( _client == null ) {
            _client = new QuittingClient( );
        }
        return _client;
    }

    public void actionPerformed((ActionEvent actionEvent) ) {
        System.exit(0);
    }
}
```

(b) [6 marks] Company X has a large amount of applications software written using a particular class library. Company Y that wrote the class library has now gone out of business. Company X buys a new class library from Company Z that provides the same functionality as the previous library, but unfortunately many of the classes have different interfaces. Company X does not have access to the source code for the old or the new library. What should Company X do??

Select the most appropriate design pattern to use to address the above problem and show how it is applied. In particular, show an appropriate class diagram(s) and enough code fragments to illustrate your use of the pattern to solve the problem.

END OF EXAM