CMPUT 301 2013 Fall Term Final Exam
TEST VERSION: U

by Abram Hindle (c) 2013
hindle1@ualberta.ca

Name:____

CCID:____

Student Number:________________________________

Question	Mark	Out of
Object Oriented Analysis: Potential Classes and Methods		2
UML: Association, Aggregation, Composition?		2
Use Cases and Use Case Diagram		2
UML Sequence Diagrams		3
Software Processes		3
Human Error and Usability		2
Human Error and User Interfaces		2
Design Patterns		3
OO Principles		2
MVC and Observer Pattern		3
Decorator Pattern and Refactoring		2
Testing		2
Refactoring		3
TOTAL (with 1 bonus mark)		30

CMPUT 301 Fall 2013 Final

Name:____

CCID:____

Object Oriented Analysis: Potential Classes and Methods [2 marks]

Read the following paragraph and **draw** a UML class diagram of this scenario. This is about the domain, the requirements, not the final design. **Label** relationships. **Highlight** the nouns that become classes with **squares**, and the verbs and relationships with **circles**. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate.

Tower defense is a kind of game where you protect an end goal (such as a castle) from a stream of enemies that approach down a path. If the enemies are not destroyed by the time they reach the end goal, the player is hurt. If too many enemies hurt the player, the game is over. I want a CMPUT301 tower defense game. The end goal in this game is a time sink (such as netflix) where time is wasted. The enemies will be “spare time”. “Spare time” will march down a path towards the “time sink”. The player seeks to suck up this spare time with towers that attack and eat the spare time. These towers include “pair programming”, “snack food”, “caffeine”, “discipline”, “Teaching Assistant”, and “project manager”. These towers attack the “spare time” and consume the hours (hours are hitpoints) into the project. You can spend these hours on more towers to protect against the spare time from reaching a time sink. If 20 hours of “spare time” reach the end goal, the project cannot be completed and the player loses the game.

CMPUT 301 Fall 2013 Final

Name:____

CCID:____

UML: Association, Aggregation, Composition? [2 marks]

Convert this Java code to a **UML class diagram**. This Java code meant to represent a multi-network instant messaging chat client**.** Draw a well-designed **UML class diagram** to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate.

```
public interface Network {
      public List<Channel> getChannels();
      public List<Server> getServers();
}
public interface Server {
      public Network getNetwork():
      public Channel joinChannel(String name);
}
public interface Channel {
      public void sendMessage(Message msg);
      public void addListener(CListener l);
}
public interface Message {
     String getMessageText();
}
```

```
public interface CListener {
     void update(Channel c);
}
class IRCNetwork implements Network {...}
class IRCServer implements Server {...}
class JabberNetwork implements Network
{...}
class JabberServer implements Server {...}
class IRCChannel implements Channel {...}
class JabberGroupChat implements
Channel {...}
class IRCMessage implements Message {...}
class JabberMessage implements
Message{...}
```

CMPUT 301 Fall 2013 Final

Name:____

CCID:____

Use Cases and Use Case Diagram [2 marks total]

What are the titles of **three** primary use cases of the following situation:

Background:

Cardiopulmonary resuscitation (CPR) skills decay rapidly and CPR certification is a lengthy and expensive process. We wanted to automate CPR instruction and certification.

Description:

I want to design a Kinect-based (depth-video sensor) CPR training program that instructs users on proper CPR techniques; monitors and scores their CPR applied on a CPR dummy; and allows CPR videos to be forwarded to instructors to inspect the results and award certificates. Instructors need to watch the videos and decide if the recorded CPR meets the requirements of certification, provide feedback and award certificates if the CPR is acceptable

Use case 1:____

Use case 2:____

Use case 3:____

Now complete this **UML use case diagram**, including boundary, actors, use case bubbles and relationships between actors and use case.

CMPUT 301 Fall 2013 Final

Name:____

CCID:____

UML Sequence Diagrams: [3 marks]

Convert this use case sequence of steps into a **sequence diagram,** remember to include all the **actors**, the **roles**, the **components**, the **lifelines**, and **activations!** and use good names for the methods.

Use Case Sequence: Updating Fridge Tablet and getting relevant recipes.

1. **I** want to make something using the ingredients in my fridge, so on my **fridge's tablet** I click, “Update Fridge Contents”
2. **Fridge tablet** shows me the last list of fridge contents.
3. For each item in the fridge that I have consumed I remove it from the list.
4. Then I select “Search for Recipes based on Fridge Contents”
5. **Fridge tablet** shows me the ingredients that it will use in its query and then **Fridge tablet** queries the **recipe server** and gets a list of relevant recipes.
6. For each ingredient **I** don't want to use, **I** remove it from the list and **fridge tablet** will ask the **recipe server** for a new list of recipes.
7. **I** select the recipe **I** want and **Fridge Tablet** displays the recipe.
8. Once **I**'m done **I** tell **fridge tablet** to update my fridge contents to reflect the items that were consumed to make that recipe.

CMPUT 301 Fall 2013 Final

Name:____

CCID:____

Software Processes: [3 marks]

1. [1 mark] Using Git repositories **how** would you enable or help track a **staged delivery process** where clients might be using older (but maybe stable versions) of your software?

2. [1 mark] In a **daily scrum meeting** why would one use the **git log** command?

3. [1 mark] A. **Give** 1 reason **why** the Unified Process similar to the waterfall process?
B. **Give** 1 reason **why** the Unified Process is different from the waterfall process?

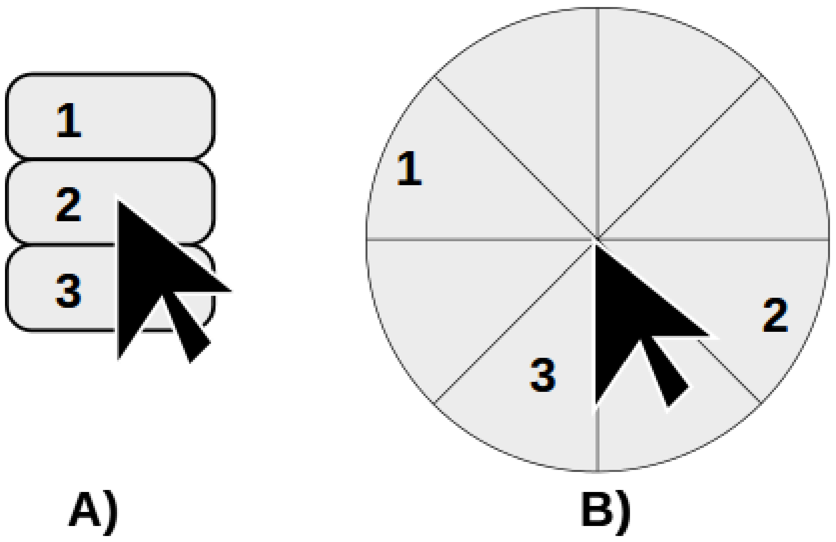
CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Human Error and Usability: [2 Marks]

[2 marks] The figure depicts stacked buttons and radial pie slice buttons. If you had to click button 1 then button 2 then button 3, use Fitt's law and Hick's law to explain **which** configuration (A or B) would be faster. Explain **why** would A or B be faster? The mouse cursor shows initial placement.



CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Human Error and User Interfaces: [2 Marks]

[1 mark] How can you design against **Saccadic Masking** causing a user to miss important information?

[1 mark] The light switches in V103 and CSC B-10 and CSC B-2 use red for on and green for off. A) **Which** subset of the population will be challenged by this configuration? B) **How** would you **redesign** these light switches?

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Design Paterns: [3 Marks]

Read the following problems, then choose and a)**NAME** the design pattern and b)**EXPLAIN** why this design pattern is the most appropriate solution.

1) You are making a city simulator. In this simulator you have modelled people. People age over time and as they age their behaviour changes, but their identity does not. How do you model the difference in behaviour between an 16 year-old teenager and a 40 year-old CEO that were the same person.

2) You're making a web interface to Eclipse, the IDE you used in the course. You want to send requests to Eclipse to open, view, modify, save, build, compile, run, cleanup, verify, checkout, etc. different projects via a web browser, yet executed in your Eclipse IDE. Your Eclipse will run a webserver to do this.

3) You're making a an operating system abstraction layer so you can port your apps between different platforms. You have defined the interfaces, but the client code still needs to get concrete instances of those interfaces. How can you build the appropriate concrete instances for the clients based on their OS?

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

OO Principles: [2 marks]

[1 Mark] **Explain** how the **hide delegate** refactoring applied to the **message chains** bad smell increases or decreases **coupling**?

[1 Mark] **Explain** how coding to the **specification** rather than the **implementation** increases or decreases **coupling**.

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

MVC and Observer Pattern: [3 Marks]

[1 Mark] **How** does the observer pattern **decouple** a model from views? Do not define model, do not define view. Tell me **HOW** this pattern works and why it **DECOUPLES**.

[2 Mark] **Draw** the **UML Sequence Diagram** for the AntHillModel's update() method in an MVC system that uses some of the following classes in the most logical way. Assume that the AntHillModel has at least 1 AntHillSideView listener.

- AntHillModel: A model of an Ant Hill.
- Ant: A model of an individual Ant in the Ant Hill.
- AntHillSideView: A view of the ant hill that shows the side view of the tunnels of an Ant Hill and shows individual ants inside the tunnels.
- Model: An abstract implementation of the observer pattern
- Listener: An abstract implementation of observable from Observer

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Decorator and Refactoring: [2 Marks]

Provide the **UML class diagram** and of DatabaseReader and its subclasses after you have refactored the read() method using **Decorator** Pattern. No sub class code is required, method names in the UML and the read method is good enough.

```
class DatabaseReader {
      ...
      Database read() {
            InputStream in = null;
            if (this.remote) {
                  in = new HttpInputStream( this.filename );
            } else {
                  in = new FileInputStream( this.filename );
            }
            if (this.compressed) {
                  in = new DecompressInputStream( in );
            }
            if (this.encrypted) {
                  in = new DecryptedInputStream( in, this.privateKey );
            }
            Database dbOut = databaseFromStream( in );
            in.close();
            return dbOut;
      }
}
```

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Testing: [2 Marks] Write the code for a **mock object class** (MockUSBConnection) that will allow testing of line **11** of **ThreeDeePrinter** in **testPrinterRetry** of **Test3DPrinter.** Write the code for **MockUSBConnection.**

```
// Prints 3D Shapes on a 3D printer in plastic
class ThreeDeePrinter {
    ThreeDeePrinter( USBConnection usbConnection ) {
        this.usbConnection = usbConnection;
    }
    boolean extrudeShape3D(Shape3D shape, int triesLeft) {
        try {
            usbConnection.restorePrinterState();
            usbConnection.send(shape);
        } catch (OutOfABSPlasticError e) {
            if (triesLeft > 0 && usbConnection.waitForReload()) {
11:             return extrudeShape3D( shape, triesLeft - 1);
            }
            return false;
        }
    }
    // waits until ABS Plastic spool is reloaded (true) returns false if cancelled
    boolean waitForReload();
    ...
}
interface USBConnection {
    void restorePrinterState();
    void send(Shape3D shape) throws OutOfABSPlasticError;
    boolean waitForReload();
}
class Test3DPrinter extends TestCase {
    void testPrinterRetry() {
        Shape3D shape = new TestShape();
        ThreeDeePrinter printer = new ThreeDeePrinter(new MockUSBConnection());
        assert(false == printer.extrudeShape3D( shape, 0));
        assert(false == printer.extrudeShape3D( shape, 1));
        assert(false == printer.extrudeShape3D( shape, 3));
    }
}
```

CMPUT 301 Fall 2013 Final;

Name:____

CCID:____

Refactoring: [3 Marks]

```
void a3d2g99(int accesslevel, String name, String address, String
email) {
     initPrinter();
     u3487dcjk2();
     initSystem();
     if (accesslevel == 1) {
          User user = new User(name, address);
          user.setEmail(email);
          emitSuperUser(user);
          sendEmail(email, “Your account has been accessed”);
     } else if (accesslevel == 2) {
          User user = new User(name, address);
          user.setEmail(email);
          emitHeadUser(user);
          sendEmail(email, “Your account has been accessed”);
     } else if (accesslevel == 3) {
          User user = new User(name, address);
          user.setEmail(email);
          emitNormalUser(user);
     }
     showDialogue(“Emitted”);
     Logger.log(name + “ emitted”);
}
```

[2 mark] **List** at least **3 bad smells** one finds, and then at least **2 refactoring** one could apply to this code snippet and then **draw** the **UML class diagram** of the relevant code after you applied these refactorings. State assumptions.