CMPUT 301 2014 Fall Term Final Exam
TEST VERSION:

by Abram Hindle (c) 2014
hindle1@ualberta.ca

Name:_____

CCID:_____

Student Number:_____

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

CMPUT 301 Fall 2014 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.

We track coyotes and other wildlife in the river valley. We use motion sensing game-cameras (hunting cameras) to track wildlife movement through a few points along the river valley. We then take the photos and put them in a database annotated by the the animal depicted, the location, the time and date. We have problems with this approach because there are a lot of photos. Annotating the photos is annoying and typing in the date, time and location is redundant because we know which device it came from. We want a system that will automatically tag photos with this data and try to identify the animals present in a photo.

CMPUT 301 Fall 2014 Final

Name:_____

CCID:_____

UML: Association, Aggregation, Composition? [3 marks]

Convert this Java code to a **UML class diagram**. This Java code meant to represent a 3D surface editor system**.** 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 Vertex {
    public double getX();
    public double getY();
    public double getZ();
}
public interface Edge {
   public Vertex[] getVertices();
}
public interface Face {
   public Edge[] getEdges();
   public Normal getNormal();
}
public class Normal implements Vertex {
   ...
}
```

```
public TwoFace implements Face {
    Edge[] edges;
    ...
}
public class Triangle implements Face {
   Vertex[3] vertices;
   Edge[3] edges;
   ...
}
public class UnionOfFaces implements Face
{
    Face[] faces;
    ...
}
```

CMPUT 301 Fall 2014 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:

I want to play music with others, but remotely. I want to Jam (play music) with other musicians across the internet.

Description:

I want a system where I can open a private jam session and invite other users. All of our audio will be streamed to each so we can hear each other in the jam. I also want to browse available public jam sessions or make my own. Our jam (our music) should be recorded and made available. If I have made a jam session I should be able to kick out members I don't like, but they can't kick me out. The administrator should be able to delete jam sessions that use copyrighted material, incase of a DMCA notice.

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 2014 Midterm

Name:_____

CCID:_____

Use Case: [3 marks]

Convert this scenario or part of it into a single **use case** related to updating Student Picker with more evaluation features. **R**emember to include of all the actors. And cover common **exceptions**. You can use the back of the page if you need space.

Scenario: Getting student picker to record

I want to randomly choose a student from the class, and ask them a question. **I** click on Student Picker and it chooses a student to question. **I** call out the student's name and if no one responds **I** indicate to Student Picker that there was an unacceptable response. Student Picker records this information and then chooses another name. **I** call out to the new student and ask them the question, they are in attendance and they respond. **I** indicate to student picker that **I** am satisfied with the response and it records that the student was successful. **I** repeat this as necessary. Once **I**'m done a class, **I** ask student picker to email me a report of students who answered and did not answer the question.

Use Case Name:

Basic Flow (back page use is OK):

Participating Actors:

Goal:

Trigger

Precondition:

Postcondition:

Exceptions (back page use is OK):

CMPUT 301 Fall 2014 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: Trading Ore in Space Spreadsheets

1. I post an ad at the local space station saying I wish to trade Ore
2. A trader submits an offer to me through the space station. The offer contains price and quantity of Ore wanted.
3. I receive the offer from the space station and decide to accept or decline.
4. I indicate accept, the station takes the ore and gives me the trader's money.
5. The ad is removed.

Exceptions:

4.1 I decline, the trader is notified. No ore or money is transfered.

CMPUT 301 Fall 2014 Final

Name:_____

CCID:_____

Software Processes: [3 marks]

[1 mark] In what parts of the waterfall model would you use refactoring?

[1 mark] How does the use of version control, like **git,** relate to the notion of courage in agile software development?

[1 mark] How is Test Driven Development employed in the design of APIs?

CMPUT 301 Fall 2014 Final;

Name:_____

CCID:_____

Human Error and User Interfaces: [2 Marks]

[2 mark total] The image above is of a toolbar for a program where it you can choose which mouse-based tool to use. Edit objects or Delete Objects. By clicking on Edit you will be able to click on objects to edit them, by clicking on delete you can click on objects to delete them.

1. What kind of common error will the user interface cause the user (what is the name of the error).

2. How do you fix it?

CMPUT 301 Fall 2014 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 have an image gallery program, the images are very large. They take a long time to load, so you generate thumbnails to represent the images, until they are needed or loaded. You might not need to load all of the images.

2) You are writing files to disk. Some files should be encrypted. Some files should be compressed. Some files should be compressed and encrypted.

3) You're making a multiplatform user interface library. Developers should be able to ask for widgets and buttons and get the appropriate one for their platform, without having to know what actual button or widget they get.

CMPUT 301 Fall 2014 Final;

Name:_____

CCID:_____

OO Principles: [2 marks]

[1 Mark] Compare how Java's **dot operator** (virtual dispatch, obj.methodCall()) and the **switch statement** increase or decrease **coupling**?

[1 Mark] **Explain** using why using an ArrayList as a Stack violates the Liskov substitution principle.
e.g. class Stack extends ArrayList<Object> { }

CMPUT 301 Fall 2014 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 observer pattern when the model has been changed. In your sequence diagram show how an abstract model instance will update all of the listening views.

CMPUT 301 Fall 2014 Final;

Name:_____

CCID:_____

Template Method, Factory Method and Refactoring: [2 Marks]

Provide the **UML class diagram** and of ImageReader and its subclasses after you have refactored the read() method using the **Template Method** Pattern and **Factory Method** Patterns. No sub class code is required, method names in the UML and an implementation of the **read** method is good enough.

```
class ImageReader {
      ...
      Image read(String filenameOrURL) {
            InputStream in = null;
            if (isURL(filenameOrURL)) {
                  in = new HttpInputStream( this.filename );
            } else if (filenameOrURL.equals("STDIN")) {
                  in = System.in;
            } else if (isFile(filenameOrURL)) {
                  in = new FileInputStream( this.filename );
            } else {
                  throw new InvalidFileSpecException();
            }
            Image image = Image.imageFromStream( in );
            return image;
      }
}
```

CMPUT 301 Fall 2014 Final;

Name:_____

CCID:_____

Testing: [2 Marks] Many GPS devices will disable themselves temporarily if they detect they are going faster than 300 km/h. The reason is that the US government doesn't want people to use GPS for missiles and other military ordinances. Write the code for a **mock object class** (**MockLocation**) that will allow testing of line **8** of **PersonTracker** in **testTooFast** of **TestPersonTracker.** Write the code for **MockLocation.**

```
class PersonTracker {
    Location lastLocation = null;
    // called every second
    void updateLocation(Location l) throws GPSException {
           if (lastLocation!=null) {
               if (lastLocation.distance(l) > 300.0*1000/3600) { // 300km/h in m/s
                  GPS.getGPS().disable(300);
8:                throw new GPSException("Too Fast!");
               }
           }
           lastLocation = l;
    }
}
// distance in meters;
interface Location { public double distance(Location l); }
class TestPersonTracker extends TestCase {
      void testTooFast() {
            PersonTracker p = new PersonTracker();
            Location l = new MockLocation(0,0);
            try {
                p.updateLocation(l);
                p.updateLocation(l);
                assert(false, "This was supposed to fail");
            } catch (GPSException e) {
                return; // we succeeded
            }
     }
}
```