CMPUT 301 Winter 2013 Final

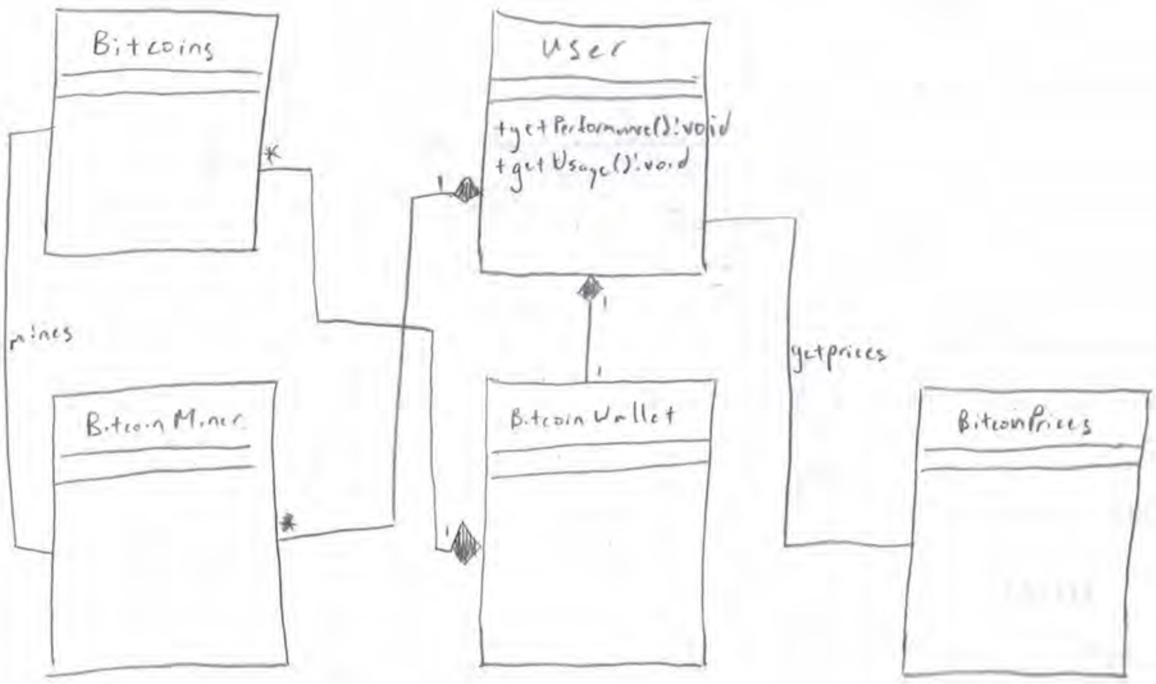
Name:________________________________________

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.

BitCoins are an alternative form of digital currency that provide for anonymous transactions. BitCoins can be mined. One problem with mining bitcoins is that you need to manage many computers (BitCoin Miners) that are dedicated to mining bitcoins. It is important to monitor their power usage, and their bitcoin output. As well, we need to monitor the historical performance of a BitCoin Miner so that if we see that its cost in terms of power is more than its benefit in terms of bitcoins. If a BitCoiner miner mines a BitCoin it must go into our BitCoin wallet. BitCoins tend be be priced by the market so we need to integrate a service that provides BitCoin prices.



CMPUT 301 Winter 2013 Final

Name:__________________________________________________

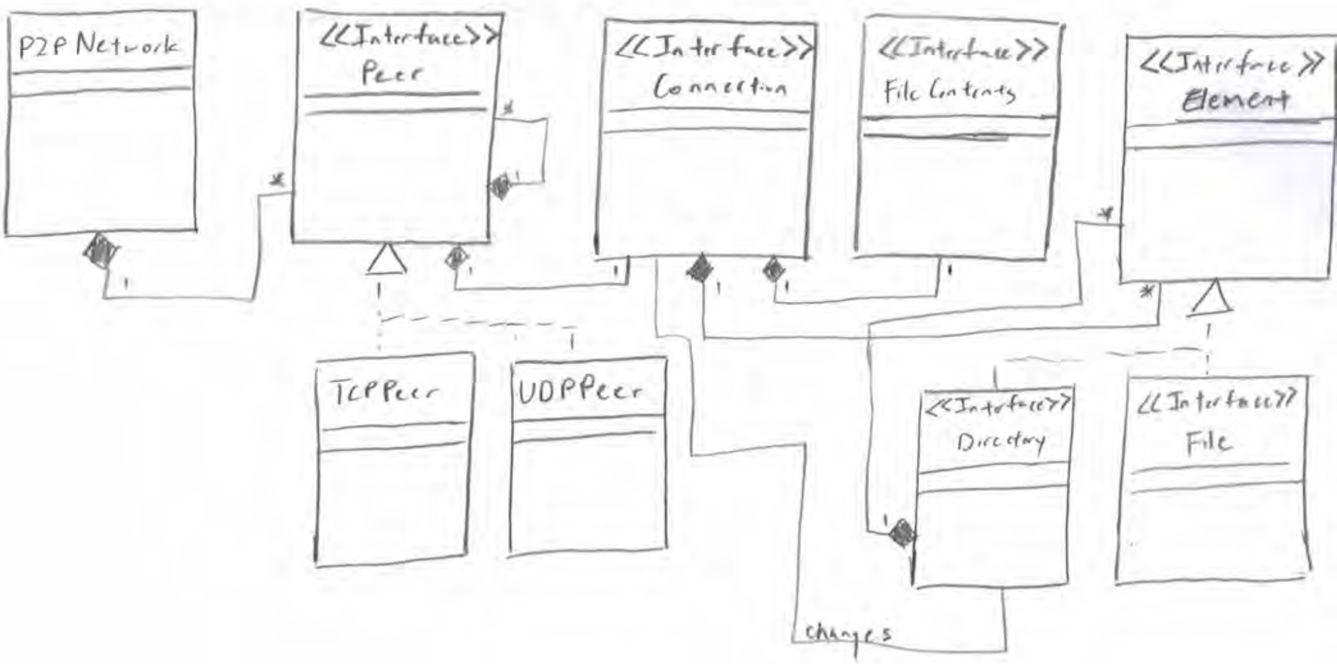
CCID:__________________________________________________

UML: Composition or Aggregation? [3 marks]

Convert this Java code to a **UML class diagram**. This Java code meant to represent a **Peer-to-Peer (P2P) file sharing network.** Draw a well-designed **UML class diagram** to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate.

```
public class P2PNetwork {
   String getName();
   List<Peer> getPeers();
}
public interface Peer {
   Connection connect();
   List<Peer> neighbors();
}
public interface Connection {
   void changeDir(Directory d);
   List<Element> listDir();
   FileContents download(File f);
}
```

```
public class TCPPeer implements Peer
{...}
public class UDPPeer implements Peer
{...}
public interface FileContents  {...}
public interface Element {
   boolean isComposite();
}
public interface File implements Element
{...}
public interface Directory implements
Element {
  List<Element> getChildren();
}
```



CMPUT 301 Winter 2013 Final

Name:____________________

CCID:____________________

Use Cases and Use Case Diagram [2 marks total]

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

**Background:**

The beauty of opensource software is that anyone can be paid to implement features for opensource projects. I want to enable crowd-funding of features.

**Description:**

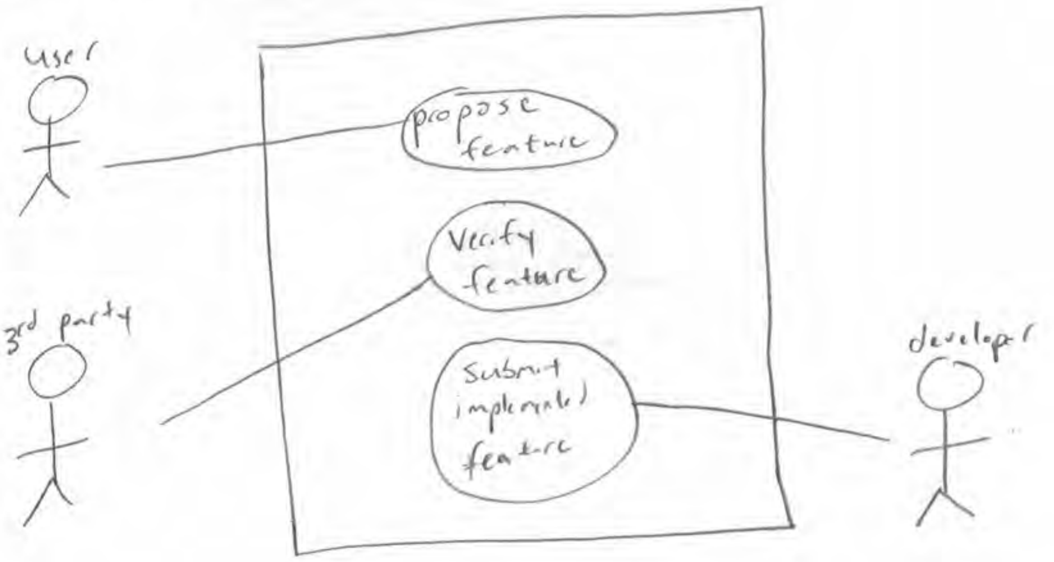
I want to design and post an feature to be built for a project, I will also specify how much I will pay for this feature. If someone wants to develop this feature they will see the posting, implement the feature, and then attach the results to my posting. A paid third party will take 1% of the posting fee and verify if the feature has been completed. If so the bounty will be paid off to the developer, and the poster (me) will be notified that the feature is ready.

**Use case 1:** user proposes feature

**Use case 2:** third party verifies feature

**Use case 3:** developer submits implemented feature

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



CMPUT 301 Winter 2013 Final

Name:______________________________________________

CCID:______________________________________________

UML State Diagram [3 marks total]

Your unimaginative boss is making you code a videogame like Super Mario: **Alright Alan**. In **Alright Alan**, **Alan** explores an office environment, **Alan** has 3 tries (lives) to navigate the office to get home. Alan starts off short as Small Alan. If an enemy, a co-worker or his boss, manages to grab **Alan**, **Alan** will be forced to stay late and will lose a try (Caught Alan). But Alan can collect powerups which help him avoid work!

- If **Alright Alan** collects a **TPS-report** he is invicible for 10 seconds and cannot be grabbed by an enemy. After 10 seconds, **Alan** will return to his previous state. (Invicible Alan)
- If **Alright Alan** collects a **coffee**, he grow twice as tall, and if an enemy grabs him, he will revert back to his original short size, but will not lose a try! (Caffinated Alan)
- If **Alright Alan** collects a stapler, **Alan** grows twice as tall AND he can fire staples at his coworkers, temporarily disabling them. If an enemy catches **Alright Alan** with a stapler, **Alright Alan** loses the stapler, and shrinks back to original size but will not lose a try. (Stapler Alan)

Your job is to **make** a **UML state diagram** that models Alan's **states**: Small Alan (default), Invicible Alan, Caffinated Alan, Stapler Alan, and Caught Alan (when grabbed and loses a try). Also in the **UML state diagram** be sure to show the transition between these states. Using this diagram I should be able to see how Alan transitions from Small Alan into Invicible Alan.

Refer to Winter 2014 final

CMPUT 301 Winter 2013 Final

Name:________________________________________

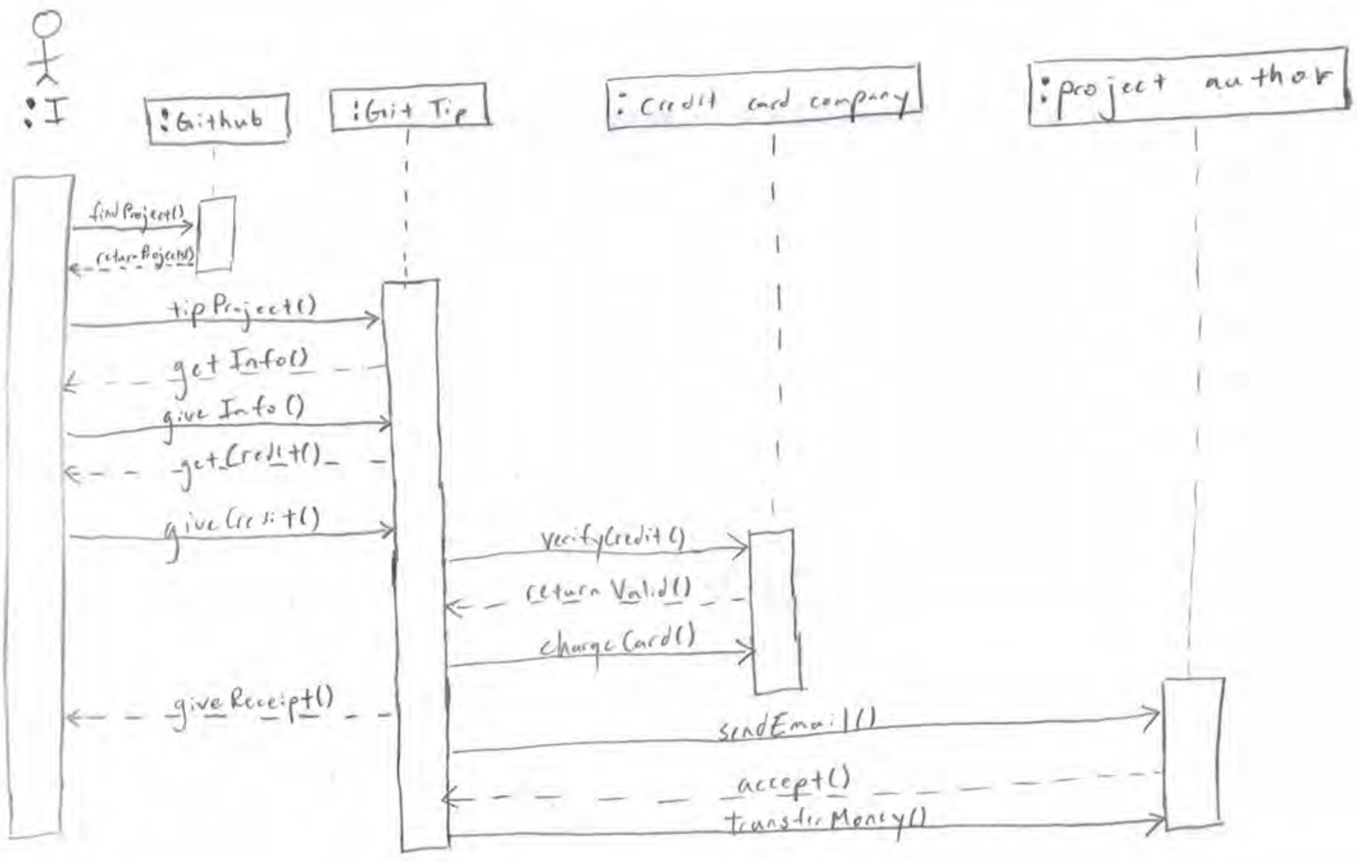
CCID:________________________________________

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: Git Tipping (see http://gittip.com/ )

1. On Github I find a project that I like
2. Then I go to Git Tip and say I want to give money this github project
3. Git Tip asks for personal information. I provide the personal information.
4. Git Tip asks for credit card information. I provide credit card information.
5. Git Tip checks with the credit card company if my information is correct.
6. Git Tip charges my card.
7. Git Tip gives me a receipt.
8. Git Tip Emails the GitHub project's author with payment information.
9. GitHub Project Author clicks accept on the email link.
10. Git Tip transfers the money to the author's account.



CMPUT 301 Winter 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?

by branching from your client's older stable version and implementing new features from there

2. [1 mark] A) **Describe** Requirements, Design and Testing stages of of the **waterfall model** and then B) **describe how** you would use Git to enable work and track work relevant to these 3 stages.

requirements is determining what the software needs to do
design is determining how we are going to implement these needs
testing is determining whether or not the coding has met these needs

I would use Git commits to track the gradual changes from each of the stages

3. [1 mark] A. **Why** would you choose a waterfall model over an iterative model? B. **Give** 1 example of an scenarios where you would use a waterfall model. C. **Give** 1 example of an scenario where you would use an iterative model.

A. You would choose waterfall over iterative model if you knew all requirements upfront

B. Waterfall would be used in a limited budget project since you would try to do things right the first time like developing a back end system for a small business

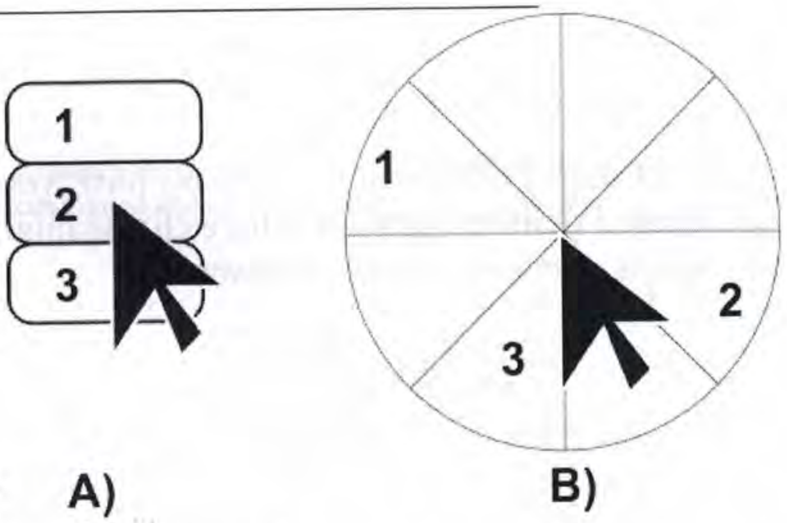
C. Iterative model would be used in projects that continually need to be improved on like developing a web browser

CMPUT 301 Winter 2013 Final;

Name:

CCID:

Human Error and Usability: [2 Marks]



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

B would be faster because there is less distance to travel between the targets for the cursor

[1 mark] A) **Why** is there **difference** in time between the choosing of 1 item from 80 unordered items and 1 item from 80 ordered items? B) **How much** approximate difference in time is there?

A) If the items are ordered you can perform binary search in your head so it's faster

B) $n - \log_2 n = 80 - \log_2 80$

CMPUT 301 Winter 2013 Final;

Name:________________________________________

CCID:________________________________________

Human Error and User Interfaces: [2 Marks]

[1 mark] How does **Saccadic Masking** affect a user scrolling through a large and long webpage using the scroll bar?

When scrolling through the webpage you may not have noticed the content has changed while scrolling because of eye movement

[1 mark] The light switches in 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?

A) People who consider green as on and red as off

B) use green for on and red for off

CMPUT 301 Winter 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're making a mind-map program, to model a web knowledge, where users can make entries that can be related to 0 or more other entries.

composite pattern because we can treat related entries uniformly

2) You're making a programmable text editor in the cloud (on the web), it can be controlled via a webpage or by other software through an API. You have a handfull of atomic operations but you want to allow automation of these operations by scripts and services. You want to compose operations together.

composite pattern because it allows for all the operations to be treated as one

3) You're making enemy characters for a videogame. The enemy characters act differently depending on if they see you, if they saw you recently, or if they are unaware of you. The enemy characters try to do their jobs when they are not aware of you. If they have seen you recently they try to find you. If they see you, they will sound an alarm and try to get you.

State pattern because the enemy behaves depending on if they see you, saw you recently or are unaware of you

CMPUT 301 Winter 2013 Final;

Name:________________________________________

CCID:________________________________________

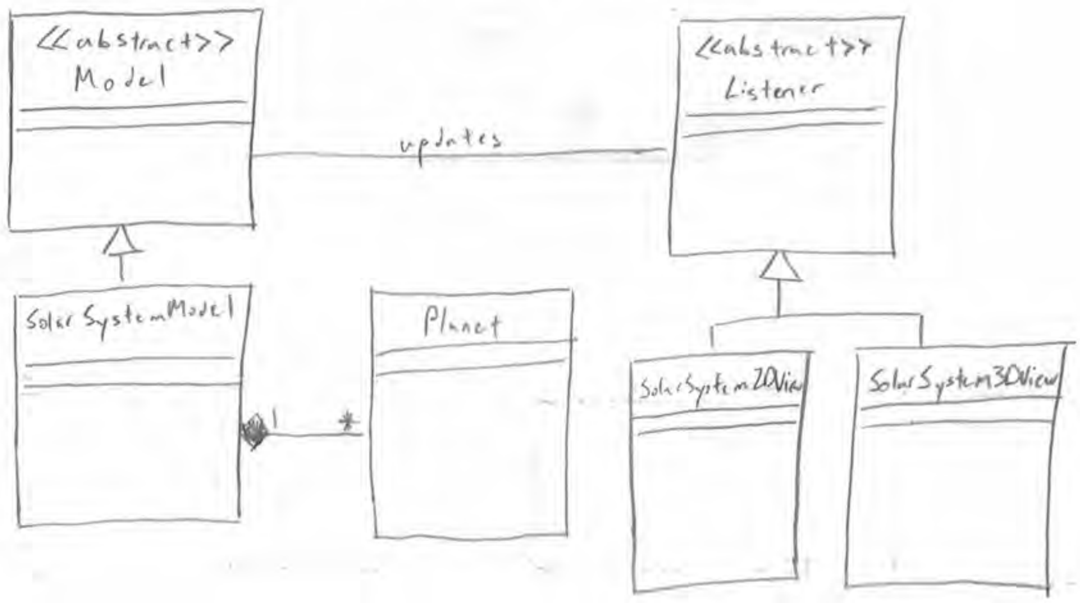
## MVC and Observer Pattern: [2 Marks]

[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**.

this pattern works by updating all views when the model is changed. It decouples because the multiple views are separated from the model. In order to add a new view it would inherit from the observer class and not affect any other classes,

[1 Mark] Draw the UML class diagram for an MVC system that uses the following classes in the most logical way. Show inheritance and relationships. Do not do methods or attributes/members.

- SolarSystemModel: A model of the solar system
- Planet: Planets within a solar system:
- SolarSystem2DView: A UI Widget that draws the Solar System in 2D
- SolarSystem3DView: A UI Widget that draws the Solar System in 3D
- Model: An abstract implementation of the observer pattern
- Listener: An abstract implementation of observable from Observer



CMPUT 301 Winter 2013 Final;

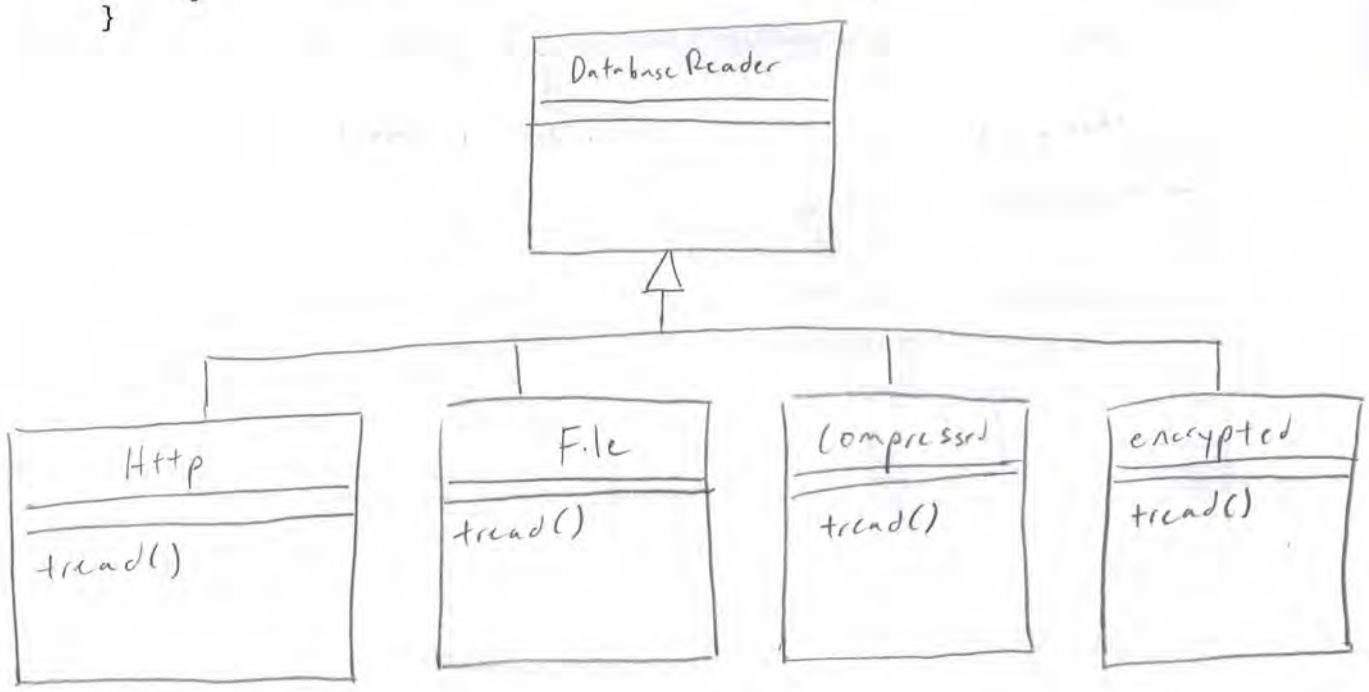
Name:________________________________________

CCID:________________________________________

## Template Method and Refactoring: [2 Marks]

Provide the UML class diagram and of DatabaseReader and its subclasses after you have refactored the read() method using Template Method Pattern. No sub class code is required, method names in the UML and the template 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 Winter 2013 Final;

Name:________________________________________

CCID:________________________________________

Testing: [3 Marks] Write the code for a **mock object class** (MockHttpClient) that will allow testing of line **9** of **ImageShare** in **testImageShareScheduleRetry** of **TestImageShare.** Write the code for **MockHttpClient**.

```
// Uploads Pictures, But my internet connection is very stable
class ImageShare {
    ImageShare(Image image, URL url, HTTPClient client) { ... };
    void upload(int retries) {
        try {
            client.httpPost( imagePostBody() );
            success = true;
        } catch (HTTPNetworkConnectionException e) {
9:          scheduleARetry(retries - 1);
        }
        return success;
    }
    void scheduleARetry(int retry);
    boolean wasSuccessful();
    int retriesLeft();
    ...
}
class TestImageShare extends TestCase {
    void testImageShareScheduleRetry() {
        ImageShare is = new ImageShare( defaultImage, defaultURL,
                                        new MockHttpClient());
        assert(false == is.upload(33));
        assert(retriesList() == 32); // 1 less retry!
    }
}
```

```
Class   MockHttpClient   extends   HTTPClient   () {

        HTTPNetworkConnectionException  e = new  HTTPConnectionException();
        throw  e;

}
```

CMPUT 301 Winter 2013 Final;

Name:______________________________

CCID:______________________________

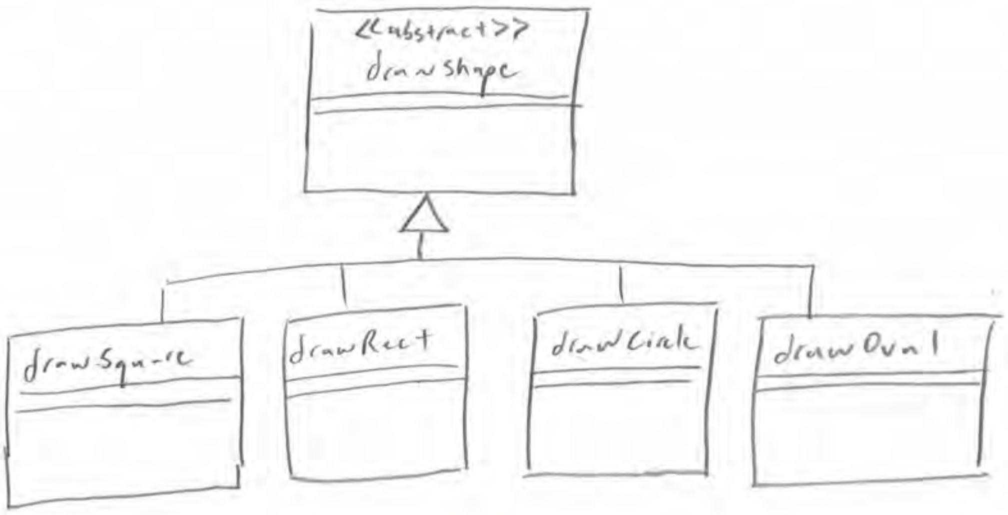
BONUS Refactoring: [3 Marks]

```
class Plotter {
     void drawHistogram(int [] counts);
     void drawLineOfValues(int [] values);
     void drawScatterPlot( int [] x, int [] y);
     void drawRectangle(int x,int y, int w, int h);
     void drawOval(int x,int y, int w, int h);
     void drawShape(int x, int y, int w, int h, int shape) {
          if (shape == 1) { //square
               drawRectangle(x,y,w,w);
          } elsif (shape == 2) { //rectangle
               drawRectangle(x,y,w,h);
          } elsif (shape == 3) { //circle
               drawOval(x,y,w,w);
          } elsif (shape == 4) { //oval
               drawOval(x,y,w,h);
          }
     }
}
```

[3 mark] **List** at least **3** bad smells one finds, and then at least **1** 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.

bad smells: blob class, data clumps, if statements

refactoring Plotter class by extracting drawShape and its subclasses



<<abstract>>
drawShape
drawSquare
drawRect
drawCircle
drawOval