CMPUT 301 2014 Fall Midterm
TEST VERSION: Pidgeot

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

Name:_____

CCID:_____

Student Number:_____

Question	Mark	Out of
Object Oriented Analysis		1
UML		3
Use Cases and Use Case Diagram		2
UML Sequence Diagram		3
UML to Code		2
Software Processes		2
MVC		2
TOTAL		15

Name:_____

CCID:_____

Object Oriented Analysis: Potential Classes and Methods [1 mark]

Read the following paragraph and pull out potential **nouns** that may lead to classes and **verbs** that may lead to relationships and methods according to Object Oriented Analysis.

I like to play my guitar. Sometimes I come up with some really neat compositions. Unfortunately I am bad at writing down the musical notes that I am playing. I want a system that listens to my guitar playing and transcribes the notes that I am playing by listening to an audio recording of the guitar. I will play notes and the system will record my notes into a MIDI file so that I can save it and I can listen to it. I also want to correct the score (the notes recorded) because sometimes my fingers fail me.

List the potential Classes [e.g. nouns]:

List the potential Actions/Methods/Relationships [e.g. verbs]:

CMPUT 301 Fall 2014 Midterm

Name:_____

CCID:_____

UML: **Composition** or **Aggregation**? [3 marks]

Convert this Java code to a **UML class diagram**. This Java code is meant to represent a worm in a worm videogame. Draw a well-designed UML class diagram to represent this information. Provide the basic abstractions, attributes, methods, relationships, multiplicities, and navigabilities as appropriate. "..." means much code is omitted.

```
public interface Worm {
  public Segment getFirstSegment();
}
class Segment {
  Segment nextSegment;
  Segment getNextSegment();
  ...
}
class EarthWorm implements Worm {
  Segment head;
  ...
}
```

```
abstract class WormHaver {
  public List<Worm> worms;
  public Worm removeWorm() { ... }
  public boolean stillHaveWorms() {...}
}
class CanOfWorms extends WormHaver
{}
class Dog extends WormHaver {}
```

CMPUT 301 Fall 2014 Midterm

Name:_____

CCID:_____

Use Cases and Use Case Diagram [2 marks total]

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

Scenario:

When I test an android application by running jUnit tests. The results of the jUnit tests are recorded. After my tests finish I can see a comparison between the current test run and prior test run. The system will inform me if I am passing more tests than before. After I commit and push to my repo the jUnit tests will be run and recorded. This will share with my team members the current status and as well if the jUnit test counts change I gain or lose points. I can see my points and the points of team members.

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:_____

Sequence Diagram: [3 marks]

Convert this use case into a **sequence diagram,** remember to include all the actors, the components, the lifelines and use good names for the methods. You can use the back of the page if you need space.

Use Case: Rock Paper Scissors

1**. Client** enters the Rock Paper Scissors (RPS) casino.
2. **Client** pays the RPS **clerk** $1 in cash (no change).
3. The **clerk** locks eyes with **Client** awkwardly and **counts down**: 3.. 2.. 1..
5. **Client** and **clerk** each choose one of 3 hand formations: rock, paper, or scissors.
6. The **clerk** evaluates who wins according to RPS rules.
6.1 The **clerk** gives the client $2 if the **client** wins.
7. The game is over
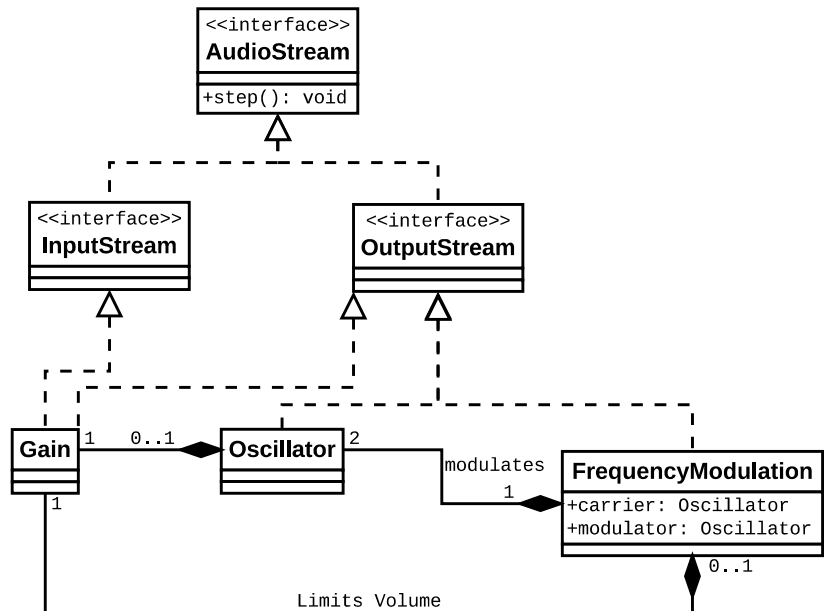7.1 If client wish to play again goto 2.

Name:_____

CCID:_____

UML to Code: [2 marks]

Convert this class diagram to
skeletal Java Code.
Include all attributes and obviously
public methods.
Includes all generalizations and
necessary associations.

```
                              <<interface>>
                               AudioStream
                              +step(): void

          <<interface>>                  <<interface>>
           InputStream                   OutputStream


   Gain  1    0..1   Oscillator  2           modulates   FrequencyModulation
                                                     1    +carrier: Oscillator
      1                                                   +modulator: Oscillator
                                                                    0..1
                        Limits Volume
```

Name:_____

CCID:_____

Software Engineering: Software Development Processes [2 marks]
Keep the responses short. A long response that is not on topic is dangerous.

[1 mark] What is an appropriate way to use **git** in order to support a staged delivery software development process?

[1 mark] List 2 concrete methods that agile methodologies use to give programmers courage (in the agile sense).

Name:_____

CCID:_____

Model View Controller Pattern [2 marks]

[1 mark] In the game Space Spreadsheets there is a view, a chart, that shows how much money the player currently has and has had during the last 10 minutes. It gets updated once a second. **Explain** why this polling every second is **different** than using the Observer pattern.
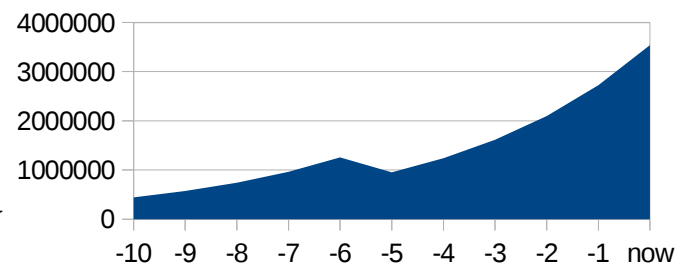
[1 mark] Provide **1 difference** and **1 similarity** between **MVC** and **3-tiered** architecture.