

CMPUT 301 2015 Fall Term Final Exam

TEST VERSION:

by Abram Hindle (c) 2015
hindle1@ualberta.ca

Name: _____

CCID: _____

Student Number: _____

Question	Mark	Out of
Object Oriented Analysis: Potential Classes and Methods		3
UML: Association, Aggregation, Composition?		3
Use Case		3
UML Sequence Diagrams		3
Human Error and User Interfaces		3
Design Patterns		3
State Pattern + State Diagram		3
Optimization and Observer Pattern		3
Testing		3
Refactoring		3
TOTAL		30

Name: _____

CCID: _____

Object Oriented Analysis: Potential Classes and Methods [3 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 are contracted to help study money laundering. We will be given the transaction history of a bank account in the form of withdrawals and deposits. We want to produce a classifier that classifies a bank account as a money laundering bank account or not. Our classifier will match the known money laundering patterns of withdrawals and deposits against the bank account under scrutiny. The classifier is extensible and can handle multiple patterns. If the bank account matches these patterns we will classify it as a laundering account.

Name: _____

CCID: _____

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

Convert this Java code to a **UML class diagram**. This Java code meant to represent an audio signal dataflow 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 Filter extends Input, Output {  
}  
public interface Process {  
    public void step();// do a unit of work  
}  
public interface Source extends Process{  
    public void connectOut(Sink sink);  
}  
public interface Sink extends Process{  
    public void connectIn(Source source);  
}  
  
public LowPass implements Filter { ... }  
public HighPass implements Filter { ... }  
public Bandpass implements Filter {  
    LowPass lowPass;  
    HighPass highPass;  
    ...  
}  
public SineOscillator implements Source {...}  
public Vocoder implements Source {  
    SineOscillator oscillators[64];  
    ...  
}
```

Name: _____

CCID: _____

Use Case: [3 marks]

Convert this scenario into a single **use case** related to organizing duplicate github bug reports/issues. Remember to include of all the actors. And cover common **exceptions**. You can use the back of the page if you need space.

Scenario: Finding duplicate bugs from Github repositories

I do this manually and we need it to be automated. I need to group bugs that are duplicates into buckets of duplicate bugs. So what I first do is find a bug that has a duplicate ID, let's say that Bug 747. Now Bug 747 says it is a duplicate of 652. So I will then query for duplicate bugs of 652 to produce a duplicate bug bucket for bug 652. Once I have those bugs I will inspect each other to ensure if it is an actual bug or not. By the time I am finished I should've joined all the duplicate bugs into separate buckets. When it is automated I would like to be asked if bugs are actually duplicates of each other.

Use Case Name:

Basic Flow (back page use is OK):

Participating Actors:

Goal:

Trigger

Precondition:

Postcondition:

Exceptions (back page use is OK):

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: Renting an AutoRental Car that you approached from the street.

1. I wipe my credit card across RFID reader mounted on the window of the car.
2. If the AutoRental Car company system has me on file they will authorize my entry to vehicle. The vehicle door will unlock.
3. I get into the car, open the glove box and insert my credit card into the keybox.
4. The keybox ejects the car key.
5. I start the car and drive around.
6. When I am done I insert the car key into the key box.
7. The system charges my credit card for the rental.
8. The keybox ejects my credit card.
9. I exit the vehicle and wipe my credit card on the RFID reader.
10. The car locks.

Exceptions:

- 3.1 if I am not in the system the car will not unlock.

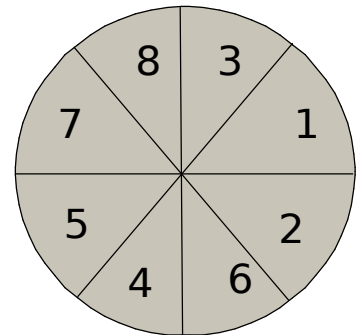
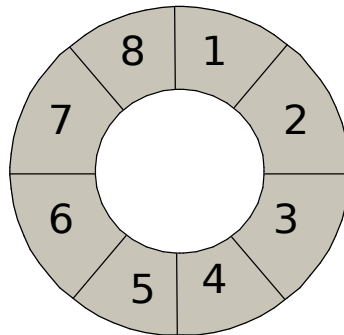
CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

Human Error and User Interfaces: [3 Marks]

Given these 2 radial menus, which one will be faster to click button 3 on? Assume that you start from the center. Your answer must be justified using Hicks Law and Fitts Law.



CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

Design Patterns: [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 want to model arithmetic expressions (such as those you could type into a scientific calculator or a graphing calculator) and get them to calculate the result in a lazy manner.

2) The user sometimes need to log all of the toString() method calls for a handful of objects of the same supertype you have made.

3) You're making a game where 3 different alien races battle against each other. To produce battle units/soldiers, the player must build a barracks for that race, which will produce only the battle units/soldiers of that race.

CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

State Pattern and State Diagram: [3 Marks]

In checkers a checker piece can move diagonally in 2 directions towards the end of the board. They may not move backwards. If a checker piece reaches the other end of the board it gets promoted to a king. Kings may move in all directions diagonally.

1. Model the state of a checker piece using a UML state diagram.
2. Draw the UML class diagram of a checker piece, and its states, that uses the state pattern that implement the states from your UML state diagram.

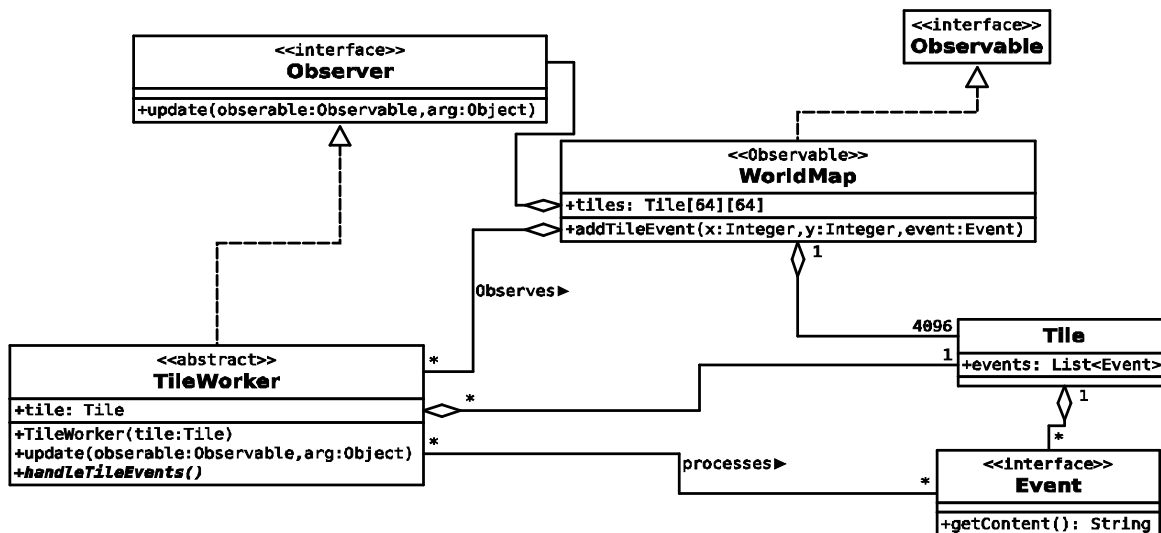
CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

Observer and Optimization: [3 Marks]

We use the observer pattern with WorldMap. We have a map represented by a matrix of tiles. And these tiles accumulate events which we want to process. We have lots of observers, these are TileWorkers, they will consume events accumulated on Tiles. TileWorkers observe the WorldMap. Tiles are shared between TileWorkers and the WorldMap. Unfortunately due to the large number of TileWorkers (1000s), notifyObserver is taking a lot of time. Using the profiler we find that most update calls are hitting irrelevant TileWorkers who have nothing to do with the updated tile. Thus we need to modify and optimize our use of the Observer pattern. **With a UML class diagram show** how you can modify this system such that TileWorkers only get updated for the Tiles they pay attention to.



CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

Testing: [3 Marks]

Write a comprehensive testcases for this clamp method code that cover all equivalence classes. Expose the bug, but make sure once fixed that other behaviour is appropriately tested.

```
class SigUtil {
    /**
        Clamp a value within a range.
        This method returns x if x is between min and max, otherwise
        if x is less than min it returns min or
        if x is greater than max it turns max
    */
    int clamp(int x, int min, min max) {
        return Math.min(max,Math.min(min,x));
    }
    ...
}
```

CMPUT 301 Fall 2015 Final;

Name: _____

CCID: _____

Refactoring: [3 Marks]

1. What is the prerequisite that needs to be met before we can refactor?

2. Name and apply an appropriate refactoring for the following, draw a UML class diagram of the resulting code and provide the refactoring name.

```
class TotalArea {
    List<Rectangle> rects;
    TotalArea(List<Rectangle> rects) {
        this.rects = rects;
    }
    int totalArea() {
        int totalArea = 0;
        for (Rectangle rect: rects) {
            int area = rect.length *
                rect.width;
            totalArea += area;
        }
        return totalArea;
    }
}

class Rectangle {
    int width;
    int length;
    Rectangle(width, length) {
        this.width = width;
        this.length = length;
    }
}
```