CMPUT 301 2015 Winter Term Final Exam
TEST VERSION: Magnetite

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?		2
UML Sequence Diagrams		2
Software Processes		3
Human Error and User Interfaces		2
Design Patterns		3
OO Principles		2
State Diagram		3
MVC and Observer Pattern		3
Testing		3
Factory Method and Refactoring		3
Testing		2
TOTAL		30

CMPUT 301 Winter 2015 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.

I am in charge of political campaign yard signs for a candidate running in the next provincial election. I need to record voter addresses of voters who want to host one of our campaign yard signs on their lawn or in their window. Then I need to get our sign delivery team to deliver a sign. After the election I will have to send out the delivery team to retrieve every lawn sign, and I'll have to phone everyone who has a window sign. I need a system to help automate recording voter addresses and keep track of where the signs are. The delivery team needs maps for distributing signs and will need a map and routes to take for picking up lawn signs.

CMPUT 301 Winter 2015 Final

Name:_____

CCID:_____

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

Convert this Java code to a **UML class diagram**. This Java code is meant to represent a video game**.** 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 StateCostume {
    public void jump(Player player);
    public void run(Player player);
}
class JumpSuit implements
StateCostume {...}
class FastSuit implements
StateCostume {...}
class SlowSuit implements
StateCostume {...}
public interface Player {
    public void jump();
    public void run();
}
```

```
public class Human implements Player
{
    Weapon [2] weapons;
    StateCostume costume;
    public void jump() { ... }
    public void run() { ... }
}
public class NPC extends Human
{ ... }
public interface Weapon {
    public void fire();
}
class Sword implements Weapon { ... }
class Rifle implements Weapon { ... }
```

CMPUT 301 Winter 2015 Final

Name:_____

CCID:_____

UML Sequence Diagrams: [2 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: Escrow (buying an item online using a middle-man (escrow) for security)

1. I find an item that I want to buy
2. I place an offer on the item
3. Seller accepts the offer
4. I place a payment in escrow (3rd party)
5. Seller ships the item to escrow (3rd party)
6. Escrow inspects the item and the payment, if acceptable Escrow ships the item to me
7. I receive the item and confirm receipt to escrow.
8. Seller receives payment from escrow.

CMPUT 301 Winter 2015 Final

Name:_____

CCID:_____

Software Processes: [3 marks]

[1 mark] How do agile software processes provide feedback to stakeholders?

[1 mark] How would one use continuous integration in staged delivery processes?

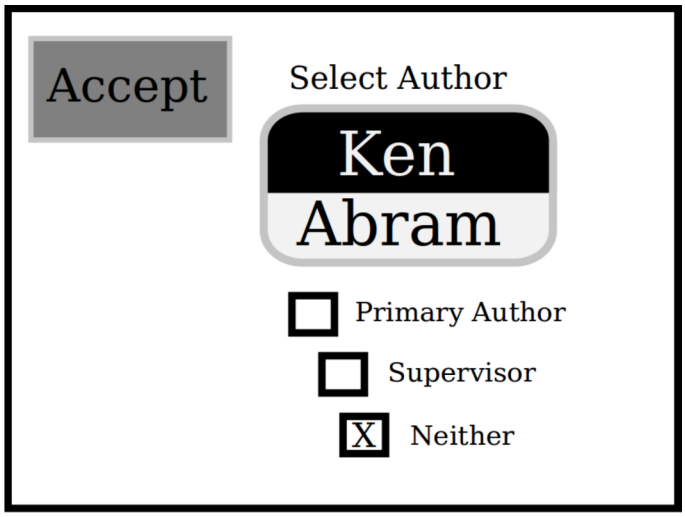
[1 mark] List 2 tools that promote the concept of courage, from agile development. Briefly explain why.

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

Human Error and User Interfaces: [2 Marks]



[2 mark] You are filtering reports written by authors. You need to indicate whether or not people in your organization, listed as authors on the paper are primary authors, supervisors or neither. This is the user interface, a dialog box that pops up when discussing a report. It contains a list of relevant authors in our organization and lets you explain the role of the each author.
List everything wrong with this user interface. Draw a better user-interface in terms of clarity, usability, and graphical design elements.

CMPUT 301 Winter 2015 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 writing database rows to disk. Depending on the database some columns have private information and need to encrypted. Some columns contain a lot of text and should be compressed. For some columns, both operations need to occur.

2) You are writing a game where you run over your enemies and they become a part of you. When you press the fire button all of the enemies that are part of you shoot in all directions.

3) You are writing a instant messaging chat client. You want the user to load 3rd party plugins that can respond to certain requests such as “where are you” or “send me your itinerary” automatically.

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

OO Principles: [2 marks]

[1 Mark] How are cohesion and coupling affected by the advice:
Depend upon abstractions. Do not depend on concrete classes

[1 Mark] Given the advice, “Favor composing objects (delegation) over implementation inheritance,” explain how cohesion and coupling are affected.

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

State Dragram: [3 marks]

Your job is to **make** a **UML state diagram** that models this simple printer described below. Also in the **UML state diagram** be sure to show the transition between these states.

The toxic receipt printer starts up when the power button is pressed. It then goes into startup mode. It determines if it has paper or not. If it doesn't have paper it goes into out of paper mode and has to be turned off to allow someone to service the printer to replace the roll of toxic paper. If the printer has toxic paper it will start to accept jobs. The printer can accept jobs at anytime after startup if it isn't out of toxic paper, even while it is printing. If the printer has 1 or more jobs it will grab the first job and print that job, removing it from the queue. Upon finishing printing the printer will determine if there is any toxic paper left. If no paper is available the printer will go to out of paper mode, otherwise it will return to servicing the job queue. If the printer has no jobs it will just sit idle until it has a job it can print.

CMPUT 301 Winter 2015 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 updates all listeners.

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

Testing: [3 Marks]

Write test-cases using jUnit style unit tests for a routine that determines if two circles intersect. Keep the number of redundant tests low but test each equivilence class.

```
class Circle {
    private double x;
    private double y;
    private double radius;
    public Circle(double radius, double x, double y) { ... }
    public boolean intersect(Circle c) { ... }
}
```

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

Factory Method Refactoring: [3 Marks]

CrossPlatformOK displays an OK Button dialogue using native widgets for Win32 and OSX, we're going to add a QT version for Linux. Before we can do this we need to apply two refactorings for CrossPlatformOK to enable the **Factory Method** pattern.
1. Provide ONLY the source code of the new show() method in CrossPlatformOK.
2. Provide the **UML class diagram** and of CrossPlatformOK and related classes after the refactoring.

```
class CrossPlatformOK {
      static int WIN32 = 1;
      static int OSX   = 2;
      String message;
      int platform;
      CrossPlatformOK(String message, int platform) {
            this.message = message;
      }
      void show() {
            if (platform == WIN32) {
                  Widget button = new Win32Button(“OK”);
                  Widget label  = new Win32Label(message);
                  Window window = new Win32DialogueWindow();
            } else {
                  Widget button = new OSXButton(“OK”);
                  Widget label  = new OSXLabel(message);
                  Window window = new OSXDialogueWindow();
            }
            window.add(label);
            window.add(button);
            window.show();
      }
}
```

CMPUT 301 Winter 2015 Final;

Name:_____

CCID:_____

Testing: [2 Marks] We have a task executor that will retry tasks that are interupted. Write the code for a **mock object class** (**MockTask**) that will allow testing of line **11** of **Processor** in **testTooManyRetries** of **TestProcessor.** Write the code for **MockTask.**

```
public interface Task { public void execute(); }
class Processor {
    void process(Collection<Task> tasks) {
        for (Task task: tasks) {
            try {
                task.execute();
            } catch( InterruptedException e) {
                try { // retry it!
                    task.execute();
                } catch (InterruptedException e) {
11:                 throw new UnfinishedTaskException(task);
                }
            }
        }
    }
}
class TestProcessor extends TestCase {
      void testTooManyRetries() {
            Processor p = new Processor();
            Task t = new MockTask();
            try {
                p.process(new ArrayList<Task>(new Task[]{t}));
                assert(false, “This was supposed to fail”);
            } catch (UnfinishedTaskException e) {
                assert(e.task == task,”Our task is not the same?”);
            }
     }
}
```