CMPUT 301 2014 Winter Midterm
TEST VERSION: S

by Abram Hindle (c) 2013
hindle1@ualberta.ca

Name:____

CCID:____

Student Number:____

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

CMPUT 301 Winter 2014 Midterm

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 want to share photos with some level of privacy. When I send a someone a photo it will be converted to grayscale, split into 20 frames that when played back very quickly (120 frames per second) exploiting the optical illusion of persistence of vision. This will stop most desktop recorders from recording the actual picture. Once the photo is viewed the user can view it 2 more times and then its key is forgotten from memory. The photoservice will also forget the key after the photo has been successfully transmitted. All photos will be encoded and encrypted then sent to the service which will distribute the photo. The photoservice will encrypt the photo so it is only for one receiver.

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

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

CMPUT 301 Winter 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 simple VideoGame with powerups. 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 Behaviour {
  public boolean invincible();
  public boolean canFly();
  public boolean canSwim();
}
class FrogSuit implements Behaviour {  ...
  public boolean canSwim() { return true; }
}
class FlyingSquirrelSuit implements
Behaviour { ... }
class IndestructibleSuit implements
Behaviour { ... }
public interface Tile { ... }
class Dirt implements Tile { ...}
class Water implements Tile { ... }
```

```
class MainCharacter { ...
  Behaviour behaviour;
  List<Weapon> weapons;
}
public interface GameContext {
  public GameMap getMap();
  public MainCharacter getPlayer();
}
public interface Weapon {
  public void fire(GameContext gc);
}
public class GameMap { ...
  Tile[64][64] tiles;
}
```

CMPUT 301 Winter 2014 Midterm

Name:____

CCID:____

Use Cases and Use Case Diagram [2 marks total]

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

Description:

When I use a bug tracker to submit bug reports I want to be warned as I write up the description of a bug if an existing bug report already exists in the bug repository. Along the side, possible matching bug reports should be shown, allowing me to browse and evaluate these reports if they already cover the problem I am reporting. Once a bug is submitted, it must be triaged, by a triager who assigns the most appropriate developer to deal with the bug report. The triager should see a list of recommended developers based on statistics from that developers history. Once an appropriate developer is found the triager assigns the bug to the developer, who will be notified that a bug has been assigned to them. The developer can then view the bug.

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 Winter 2014 Midterm

Name:____

CCID:____

Use Case: [3 marks]

Convert this scenario or part of it into a single **use case** related to 3D printing. **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: 3D Printing

I go to the 3D printer store and **I** initiate a conversation with the store **clerk**, and request that they print my custom 3D rook chess piece and that is to be chemical polished (this smoothes the object out but is optional). **I** provide the **clerk** with a **USB Keydrive** containing the 3D plans for the rook that I want printed. The **clerk** reads the **USB keydrive** and sends the plans to the **3D rights database.** I made this rook myself so the the **clerk** tells me that no rights holders have made an intellectual property claim on my object and thus they can print it. If a rights holder did make a claim I could pay an extra fee or leave. The **clerk** prints my rook using the **3D printer.** The **clerk** gives the **3D printed object** to the **finisher clerk**. The **finisher clerk** inserts my 3D printed rook into the chemical polisher. The first **clerk** rings up my order on the **Point of Sale Device** and provides me with the total cost of $27.66 based on the volume of the rook. I pay and the **finisher clerk** hands me the object after it is washed off.

Use Case Name:

Participating Actors:

Goal:

Trigger

Precondition:

Postcondition:

Basic Flow (back page use is OK):

Exceptions (back page use is OK):

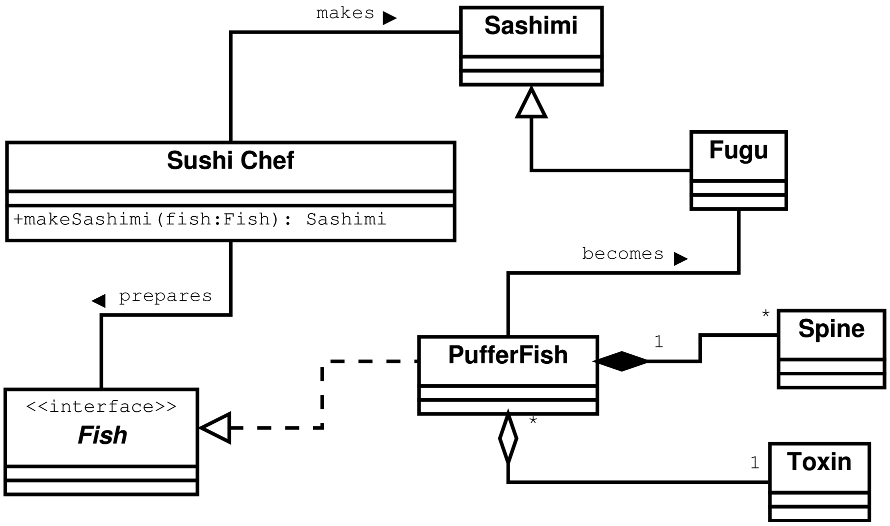
CMPUT 301 Winter 2014 Midterm

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.



CMPUT 301 Winter 2014 Midterm

Name:____

CCID:____

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

We talked about courage, and how agile methodologies used methods to improve the courage of programmers. **Explain** how **git** gives programmers **courage** with respect to agile methodologies. [1 mark]

Explain 2 reasons why the **waterfall model** and the **unified process** differ and what is 1 reason that are they similar. [1 mark]

CMPUT 301 Winter 2014 Midterm

Name:____

CCID:____

Model View Controller Pattern [2 marks]

[1 mark] Imagine a software project that models the solar system. Given a view of the solar system as a 2D chart (**Chart2DView**) that shows the planets on a plane, and a model of the solar system (**SolarSystemModel**), ***explain*** what *knowledge* this **Chart2DView** should have of the **SolarSystemModel** and what *knowledge* the **SolarSystemModel** has of a **Chart2DView.**

[1 mark] Write the java code for the method
`public void notifyAll()` of the **SolarSystemModel** class