

# CMPUT301W20B1/EB1 Final

## Test Version: 1234567-hindle1

### 2020-04-10

(c) 2020 Abram Hindle. All rights reserved. Not for redistribution

Name: Abram Hindle  
CCID: hindle1  
Student Number: 1234567  
Favourite Pokemon: \_\_\_\_\_

By signing or writing my full name again, I hereby swear I followed the solo effort policy for this exam:

---

## 1 Instructions

1. Read this front matter!
2. This is a solo effort exam.
3. All pages beyond this page are worth 3 marks.
4. There should be 14 pages of questions. Count them. If you dont have enough contact the instructor.
5. The total marks for the exam will be: 42 scaled down to 30 for the course
6. Pages marked bonus ARE NOT BONUS. They are 3 marks.
7. Bonus questions are not optional you must do them.
8. This exam is unique to you, no one else has an exam like this. DO NOT SHARE IT
9. Dont talk to anyone except Abram or the Tas about this exam.
10. It is open book, open internet, open compiler, open IDE, open diagram tool.
11. ALL SOURCES MUST BE CITED
12. Programming language used must be Java.
13. This exam must be submitted on eclass by the specified time. No later. No late submissions. If you have a bad connection or family obligations then do the exam early.
14. You maybe post your question on eclass if and only if you need clarification.
15. Do not answer other students questions on eclass about the exam.
16. SUBMISSION FORMAT: Acceptable submission formats are: .pdf and .odt.
17. Unacceptable formats are infinite and include: JPEG, PNG, RAR, ZIP, CBR, etc.
18. If you are taking photos with a phone embed them into a PDF file.
19. I dont have the fonts you have. If you care then produce a .pdf
20. Please Use a new page for each question and draw on the page if possible.
21. Please Use a new page for each question and copy and paste the original question to the page.
22. If I cant read the PDF or I cant mark it. The exam counts as incomplete.

CMPUT 301 Winter 2014 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.

Our company specializes in geo-location based software. By geo-location we mean the current position (latitude and longitude) on Earth of the user of the software. This leads to a strange class of bugs that depend on the location of the user at the time. Some bugs only occur at certain locations. We want to augment our existing issue tracker by adding the issue-reporter's geo-location to the issue-tracker's issue report of our geo-location based software. We distribute more than 1 product that has geo-location capabilities. We want to plot a map of the geo-locations of issue reports. We also want a heat-map view of the map showing the frequency of issues based on colour.

CMPUT 301 Fall 2015 Midterm

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

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

Convert this Java code that tracks Wifi Hotspots to a **UML class diagram**. 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.

```
interface Hotspot {
    public String getName();
    public Location getLocation();
}
class PaidHotspot implements Hotspot{
    private Location location;
    ...
}
class FreeHotspot implements Hotspot {
    private Location location;
    ...
}
class MultiHotspot implements Hotspot {
    public Collection<Hotspot> hotspots;
    ...
}

interface Location {
    double getLong();
    double getLat();
    String getName();
}
class Hotspots {
    private Collection<Hotspot> hotspots;
    public Hotspot nearestHotspot(Location
here) { ... }
    ...
}
class MultiThreadedHotspots extends
Hotspots {
    Collection<WorkerThreads> threads;
    ...
}
```

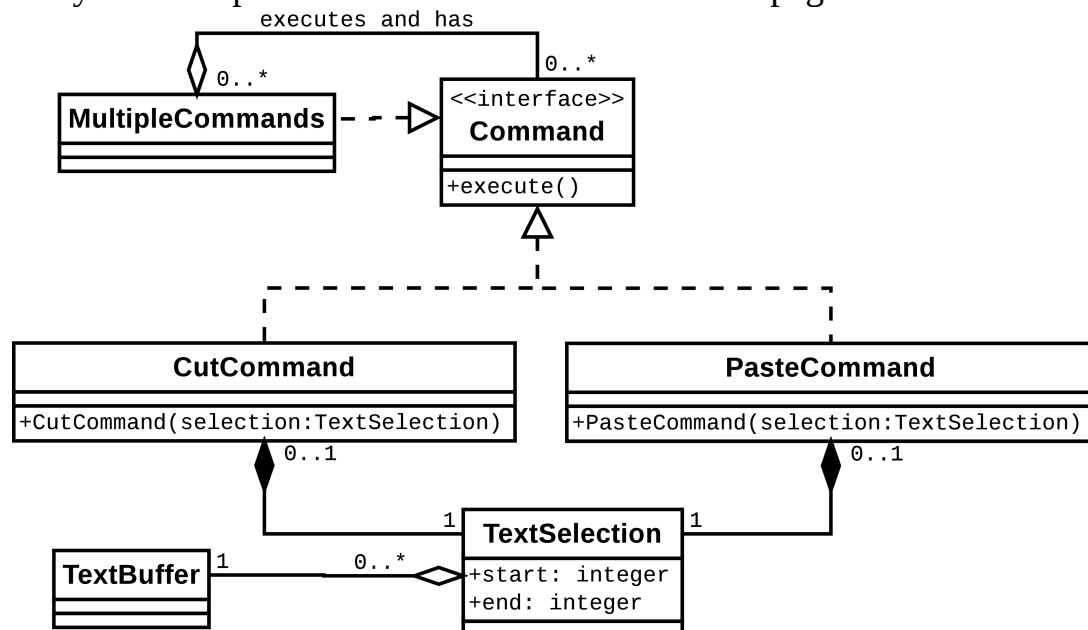
CMPUT 301 Fall 2015 Midterm

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

UML to Code: [3 marks]

Convert this class diagram of a Text Editor application to skeletal Java Code. Include all attributes and obviously public methods. Includes all generalizations and necessary associations. If you need space feel free to use the back of the page.



CMPUT 301 Fall 2015 Midterm

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):

CMPUT 301 Fall 2012 Midterm

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

Use Cases and Use Case Diagram [3 marks total]

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

**Background:**

I want to go skiing or participate in another kind of activity with my friends but we have to share some expenses. It is often difficult to get reimbursed for my expenses and it is hard to track. I want to simplify the transaction.

**Description:**

I want to design and post an activity, specify the range of people and define a budget for the activity. If someone wants to participate in this activity they must first agree to it. A participant will be required to pay a portion of the costs (divided equally among participants) and the system will take their money. The activity designer will be expected to cover all costs and thus will receive all of the budget once the activity is done.

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 2015 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: CNC Pattern Printing on Objects**

1. Client chooses the pattern on a kiosk
2. Client chooses the method of mapping the pattern to an object, on the kiosk
3. Kiosk prints a work ticket for the client, indicating the price.
4. Client gives the clerk the object to be printed on, and the work ticket.
5. The clerk places the object into the pattern printer and scans the work ticket.
6. The pattern printer prints the pattern onto the surface of the object
7. The clerk receives payment from the client
8. The clerk gives the client the object.

CMPUT 301 Winter 2014 Final

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

Software Processes: [3 marks]

[1 mark] In SCRUM what is a daily standup meeting and what are the questions asked during the standup meeting?

[1 mark] Using Git repositories **how** would you enable or help track an iterative software development process?

[1 mark] How does test first development work? How does test first development affect the design of software?



CMPUT 301 Fall 2018 Final;

Name: \_\_\_\_\_

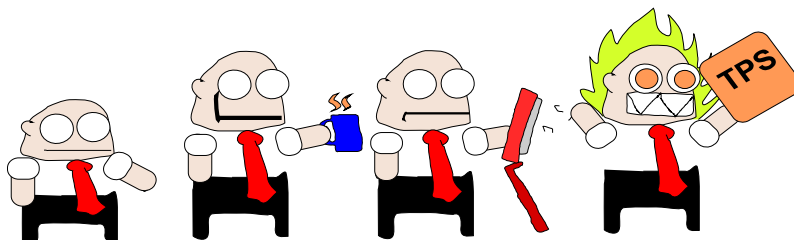
CCID: \_\_\_\_\_

Behavioural Patterns: [3 Marks]

Your unimaginative boss is making you code a videogame like Super Mario:

**Alright Alan.** In the game, **Alright Alan** explores an office environment, **Alright**

**Alan** has 3 tries (lives) to navigate the office to get home. Alan starts as Small Alan. If an enemy, a co-worker or his boss, manages to grab **Alright Alan**, **Alright 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 invincible for 10 seconds and cannot be grabbed by an enemy. After 10 seconds, **Alan** will return to his previous state. (*Invincible Alan*)
- If **Alright Alan** collects a **coffee**, he grows twice as tall, and if an enemy grabs him, he will revert back to his original short size, but will not lose a try! (*Caffeinated 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*)

1. What design pattern is appropriate for modelling Alright Alan's change of behaviour?

2. Draw the **UML class diagram** of a Alright Alan and Alright Alan's behaviour using the appropriate design pattern. Required methods are run, jump, collideWithEnemy, fireStapler.

CMPUT 301 Fall 2014 Final;

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

Testing: [2 Marks] Many GPS devices will disable themselves temporarily if they detect they are going faster than 300 km/h. The reason is that the US government doesn't want people to use GPS for missiles and other military ordinances. Write the code for a **mock object class (MockLocation)** that will allow testing of line **8** of **PersonTracker** in **testTooFast** of **TestPersonTracker**. Write the code for **MockLocation**.

```

class PersonTracker {
    Location lastLocation = null;
    // called every second
    void updateLocation(Location l) throws GPSEException {
        if (lastLocation!=null) {
            if (lastLocation.distance(l) > 300.0*1000/3600) { // 300km/h in m/s
                GPS.getGPS().disable(300);
8:         throw new GPSEException("Too Fast!");
            }
        }
        lastLocation = l;
    }
}
// distance in meters;
interface Location { public double distance(Location l); }
class TestPersonTracker extends TestCase {
    void testTooFast() {
        PersonTracker p = new PersonTracker();
        Location l = new MockLocation(0,0);
        try {
            p.updateLocation(l);
            p.updateLocation(l);
            assert(false, "This was supposed to fail");
        } catch (GPSEException e) {
            return; // we succeeded
        }
    }
}

```

CMPUT 301 Fall 2016 Final;

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

## Refactoring: [3 Marks]

1. How can continuous integration help refactoring?

2. Name an appropriate refactoring to apply for the following, then draw a UML class diagram of the resulting classes.

```
class Square {
    int width;
    int length;
    Square(int width) {
        this.width = width;
        this.length = width;
    }
    int totalArea() {
        return this.width * this.length;
    }
    int perimeter() {
        return new
            Rectangle(width,width).perimeter();
    }
}
```

```
class Rectangle {
    int width;
    int length;
    Rectangle(int width, int length) {
        this.width = width;
        this.length = length;
    }
    int totalArea() {
        return this.width * this.length;
    }
    int perimeter() {
        return 2*width + 2*length;
    }
}
```

CMPUT 301 Winter 2014 Final;

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

OO Principles: [2 marks]

[1 Mark] **Explain** how the **replace conditional with polymorphism** refactoring applied to the **switch statement** bad smell increases or decreases **coupling**?

[1 Mark] **Explain** how coding to the **specification** rather than the **implementation** increases or decreases **coupling**.

CMPUT 301 Fall 2013 Final;

Name: \_\_\_\_\_

CCID: \_\_\_\_\_

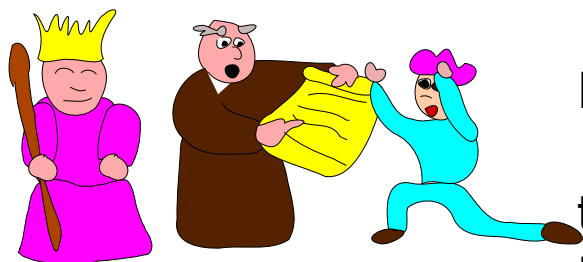
### MVC and Observer Pattern: [3 Marks]

[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 AntHillModel's update() method in an MVC system that uses some of the following classes in the most logical way. Assume that the AntHillModel has at least 1 AntHillSideView listener.

- AntHillModel: A model of an Ant Hill.
- Ant: A model of an individual Ant in the Ant Hill.
- AntHillSideView: A view of the ant hill that shows the side view of the tunnels of an Ant Hill and shows individual ants inside the tunnels.
- Model: An abstract implementation of the observer pattern
- Listener: An abstract implementation of observable from Observer

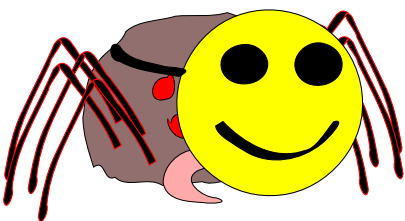
Which design pattern best describes the description and quote?



Provide a surrogate or placeholder for another object to control access to it.

*"I get in the way"*

Which pattern am I: \_\_\_\_\_



*"I'm coverup for rough interfaces"*

Which pattern am I: \_\_\_\_\_

Allow an object to alter its behaviour when its internal state changes.

*"I invert control and make others responsible for themselves"*

Which pattern am I: \_\_\_\_\_

Avoid coupling the sender of a request to its receiver by giving more than one object a chance to handle the request. *"Someone is going to deal with this, but it isn't going to be me!"*

Which pattern am I: \_\_\_\_\_



Attach additional Responsibilities to an object dynamically.

*"Sometimes I just need to look good"*

Which pattern am I: \_\_\_\_\_

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.

