

CMPUT 301 2015 Fall Midterm

TEST VERSION:

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

Name: _____

CCID: _____

Student Number: _____

Question	Mark	Out of
Object Oriented Analysis		1
UML		3
Use Cases		3
UML to Code		3
MVC		3
Version Control		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.

We need an automated ranked-list voting machine. The machine must accept a list of candidates and allow a voter to order the list by preference. It will have a touch screen interface enabling the user to order candidates by dragging them around. Every voter will authenticate with a QR-code that is scanned by the machine. Upon voting the machine will print out a verifiable hash that can be used to later to ensure that the voters vote was registered.

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

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

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;
    ...
}
```

Name: _____

CCID: _____

Use Cases: [3 marks]

Convert this scenario or part of it into a single **use case** related to 3D/CNC printing/plotting. Remember to include of all the actors. And cover common **exceptions**. You can use the back of the page if you need space.

Scenario: CNC Pattern Printing on Objects

I walk up to the pattern printer kiosk. I select a pattern I want, Elven Vines, just like those in Rivendell where Elrond lives from Lord of the Rings. Then I place my object, a thick dowel 1 meter in length, into the CNC printer and it scans the shape of my object. Then I preview the object, but the pattern is distorted. I select the correct method of mapping my selected pattern onto the object – I choose cylindrical geometry because it best matches the object I provided. Then I select ESTIMATE and the kiosk tells me it will cost \$20. I select ORDER and the kiosk's CNC printer cuts at my dowel producing a beautiful vine-like pattern on it. After cutting my dowel, I take it and the kiosk prints an invoice for me. I will give this invoice to a cashier and will pay the cost to the cashier.

Use Case Name:

Basic Flow (back page use is OK):

Participating Actors:

Goal:

Trigger

Precondition:

Postcondition:

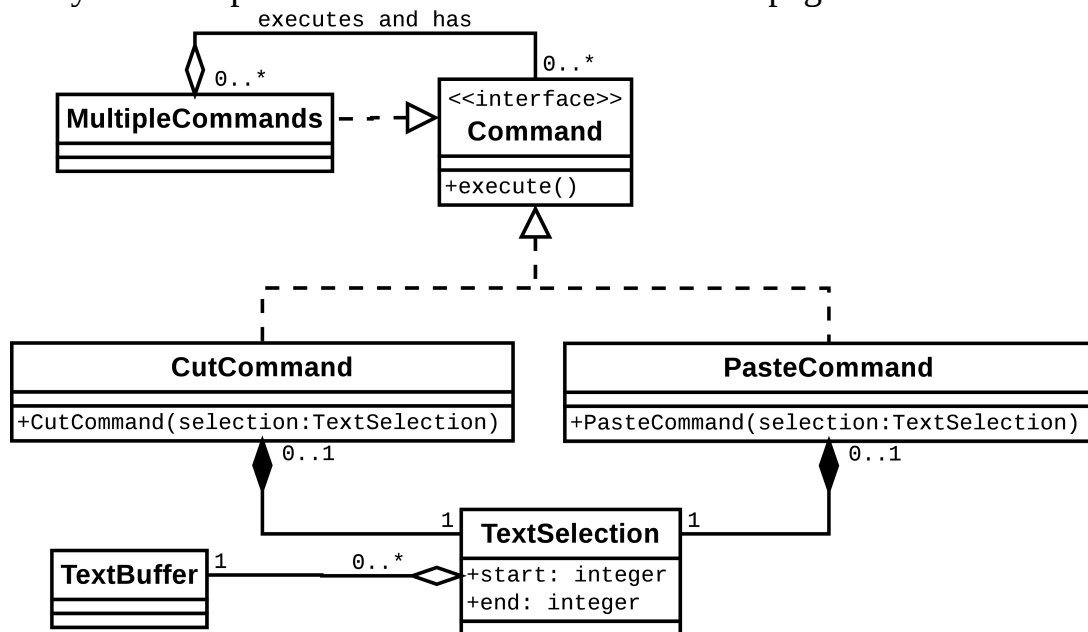
Exceptions (back page use is OK):

Name: _____

CCID: _____

UML 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.



Name: _____

CCID: _____

Model View Controller (MVC) [3 marks]

Given the following classes write down beside the class if they should have a role of Model, View, Controller, or None.

OKDialogBox
+displayConfirmation(): boolean

AddButtonClickListener
+onClick(event): void

InMemoryDatabase
-index: Index
+InMemoryDatabase(size: integer)
+getIndex(): Index

CarBattery
-charge: integer
-lastCharged: Date
+getCharge(): integer

MiniMap
-map: GameMap
-minimapView: Image
+clickListener: Listener
+update()
+setClickListener(listener)

GameMap
-map: Tile[][]
+listeners: Collection<Observer>
+notifyAll()
+getTile(point: Point): Tile

Name: _____

CCID: _____

Version Control [2] Marks

Your task is to address a serious performance issue in the project. This performance problem will require modifying 25% to 50% of the classes in the project. Furthermore the solution your team has agreed upon has not been tested and might not result in an increase in performance. Your team has delegated you responsible: you are to make these changes and prove if the changes improve performance or not. In either case your work should be isolated and separate from the rest of the team – your work should not be on *master* but your team wants to monitor your progress so you must share the commits on github.

1. Describe using git command line commands how you first isolate yourself from *master*.

2. Now you have made some initial commits, how do you share those commits to github without affecting your team?

3. Your change worked! Congratulations, performance has been improved. How do you add your changes to master?