

Midterm 2 (March 1)

THIS IS A REAL MIDTERM! This is the same structure as the practice midterm, but this midterm will have 3 questions.

The Midterm Exams are 3 exams consisting of 3 questions of 15 minutes each including upload time. You have a 5 minute grace period to upload answers to the question, and you must answer them in order.

The midterm exams are closed book solo effort. No collaboration is allowed. You will be expected to complete a question in 15 minutes, before moving on to the next question. You have a 5 minute grace period after the 15 minutes to upload that question.

In taking this exam, you agree to abide by the university policies on academic integrity. The exam is technically 50 minutes because each question will have a 5 minute grace period for submission.

You may have 2 sheets US Letter size or A4 paper with hand written notes on it. This is a synchronized midterm and it is **closed book**, but you may use 2 page paper cheatsheet that you made yourself.

Identify yourself by your CCID in each answer (a note on a diagram, a comment on a text/java file).

There are 3 questions (only 1 question for practice midterm). Each question has its own submission page:

* Question1 10:00am MDT March 1 10:00 to 10:15am MDT February 1 (10:15 to 10:20 grace period for submission)

<https://eclass.srv.ualberta.ca/mod/assign/view.php?id=5046461>

** This link will not open until the question is open

* Question2 10:15am MDT March 1 to 10:30am MDT February 1 (10:30 to 10:35 grace period for submission)

<https://eclass.srv.ualberta.ca/mod/assign/view.php?id=5046462>

** This link will not open until the question is open

* Question3 10:30am MDT March 1 to 10:45am MDT February 1 (10:45 to 10:50 grace period for submission)

<https://eclass.srv.ualberta.ca/mod/assign/view.php?id=5046463>

** This link will not open until the question is open

* If your submission is more than 5 minutes late you (outside of grace period) you must submit an explanation using this assignment. Use the textbox.

* If you have accommodations they will already be added for you, you will have more than your time modifier.

Technical Issues: Submitting to this page/assignment is to indicate technical difficulties you experienced, if we have to discuss them later. If you experience an internet outage or something similar that prevents submission you submit a response to this page.

If you miss the deadline for a question **submit anyway** and use this page to explain why your submission is late and why I should mark it. Late submissions will be accepted by eclass but reviewed at instructor's discretion.

You do not need to submit this page unless you have technical difficulties.

Each question on the midterm is marked as follows, examples are in []:

Excellent: 5 marks: Student's answer is correct, without flaw. Answer demonstrates ability to synthesize material at various levels of abstraction. Breadth and Depth of material is demonstrated and good judgement is used in providing the answer. [Perfect UML w/ 1 missing multiplicity] [Java code with 1 missing argument type]

Good: 4 marks: Like excellent but something is minor is impacted or missing. Not as consistent as excellent. [Perfect UML w/ 1 missing minor relationship] [Java code missed an exception]

Satisfactory: 3 marks Demonstrates they UNDERSTAND the core material but not subtleties. Can apply what's learned in class to simple examples or parts of the exam. There are issues with the answer but understanding of the question and answer are clear. [UML exists but is off] [Java code gets to the point but has problems]

Unsatisfactory: 2 marks. Some understanding is demonstrated but the answer is not satisfactory and is lacking. [UML exists but is not really correct] [Java code shows evidence of knowing what the problem and what the solution is but isn't getting there]

Fail: 0 marks. The student didn't understand the question and gave an irrelevant answer. The student gives no answer. The answer is not sufficient to demonstrate understanding. [UML exists but does do what is asked] [Wrong UML diagram] [Java code is not java at all]

Midterm 2 Question 1

Submit 1 PDF, PNG, or JPEG only.

5 marks.

You must add your CCID to your answer

Coupling and Cohesion

Interpret the Java code below, consider which class has poor measures of coupling and cohesion.

Refactor/restructure this code **AS A UML CLASS DIAGRAM OF RESTRUCTURED CODE** to make the least cohesive class much more cohesive, and improve its coupling. The improvement in cohesion should be significant.

Improvement doesn't mean increase, it depends on the measure we're talking about.

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 Font { ... } // Represent a font
interface Point { ... } // Represent font size
```

```

interface Country { ... } // Represent a
Country
/* Shape represents a shape with a name
 * that is intended to be drawn on the
 * shape
 */
interface Shape {
    String name();
    ...
}
/* User tracks information about a user
 */
class User {
    Font font;           // User's favourite font
for display
    Point fsize;       // User's favourite font
size
    String name;       // User's name
    String street;    // street of residence
    String city;      // city of residence
    String state;     // Or province
    Country country; // Country
    ...
}
/* A surface that can be drawn upon with text
and shapes
 */
class Canvas {
    /* drawNamedShape draws a shape s and the

```

```

shape s' name.
    * User u provides font information for
    * drawing shape s' name
    */
void drawNamedShape(Shape s, User u) {
    draw(s);
    drawText(s, u.font, u.fsize);
}
/* drawText draws the text on a Shape s
using a
    * specific Font font of Point size.
    */
void drawText(Shape s, Font font, Point
size) { ... }
/* draw a Shape s
    */
void draw(Shape s) { ... }
}

```

Midterm 2 Question 2

Plain text preferred: .txt

Submit 1 .txt, PDF, PNG, or JPEG only.

5 marks.

You must add your CCID to your answer

CRC Cards

Engage in Object Oriented Analysis and read the following scenario paragraph and pull out potential nouns that may lead to classes and verbs may lead to methods or relationships.

Make CRC cards to model each meaningful class relevant to the problem and scenario.

Responsibilities must be described and use only 1 responsibility per line.

You may use plain text, separate the cards by a blank line and the following format is recommended:

Class: X

Collaborator: Y

Responsibility: Z

...

Responsibility: Q

Scenario:

I am writing software.

I have completed adding a new feature and now I commit my changes to my source code files.

My commit is on my feature branch.

My commit is recorded locally and then I push it to the remote server to share with my colleagues.

I checkout the main branch from the remote server, then I merge my feature branch with the main branch.

Once I resolve conflicts in the source code due to the merge, I commit the merge and the fix.

Then I push my merge and fix commits to the remote server.

Midterm 2 Question 3

Submit 1 .txt, .PDF, PNG, or JPEG only.

* plain text .txt preferred *

The answers for this part are plain text. Please upload only 1 file containing your CCID and your answers to each question. An example of the answer format is at the end of this question.

5 marks.

You must add your CCID to your answer

* Process and Version Control *

We are using a process based on staged delivery.

We've already released the release called `MrMime`.

`MrMime` is released and deployed to the customer base.

The current release is called `Pichu`. This release is in its validation and testing phase.

`Pichu`'s requirements phase was started when `MrMime` was in testing.

The next release after is called Pikachu. This release is in its implementation phase.

`Pikachu`'s requirements phase was started when `Pichu` was in implementation.

The following release is called Raichu. This release is in its requirements.

`Raichu`'s requirements phase was started when `Pikachu` was 1/4 through implementation.

All releases are under version control using Git.

We have a central git repo on bitbucket (like github).

You have made a feature branch named after your CCID (Abram's feature branch is hindle1, your's is your CCID), where you iteratively prototyped a new method of saving to disk effeciently.

Your manager loves the feature and wants you to integrate it into `Pichu` and `Raichu` releases.

Assume you are working on your computer, not on bitbucket and you will have to remotely access the origin repo on bitbucket.

* If you don't make the state of the repository clear to me in your answers I won't assume it is in the correct state *

* State your assumptions *

* `...` can be used if you want to pass the time (like development time or commands) *

1. Within a single git repository hosted on bitbucket what is the best way and feature we can use to maintain staged delivery releases. For each subquestion below, name the concept, and then show the commandline git commands you would use to start each release so

that
progress on that release can be tracked.

Q1.1 Given the context that MrMime exists, how do you initialize the Pichu release? Demonstrate with the git commandline.

Q1.2 Given the context that Pichu exists, how do you initialize the Pikachu release? Demonstrate with the git commandline.

Q1.3 Given the context that Pikachu exists, how do you initialize the Raichu release? Demonstrate with the git commandline.

2. Context: You made your CCID feature branch and you based it off of

an earlier part of the implementation phase of the 'Pikachu' release.
Demonstrate how you complete each action using commandline git commands.

Q2.1 How to integrate your CCID feature with the Raichu release? Demonstrate with the git commandline.

Q2.2 How to integrate your CCID feature with the Pichu release? Demonstrate with the git commandline.

Q2.3 What is the likely complication to arise from integrating your CCID feature with the Pichu release?

Q2.4 In staged delivery would you integrate your CCID feature with the MrMime release? Demonstrate with the git commandline.

3. Context: You're working on Pikachu and you want to grab all the

latest changes from Pichu.

Demonstrate how you complete each action using commandline git commands.

Q3.1 How to integrate recent changes from Pichu into Pikachu?
Demonstrate with the git commandline.

Example suggested answer format (these aren't the answers):

CCID: hindle1

Q1.1: I love cheese

Q1.2: Brie is great

...

Sure is.

Q1.3: Gouda is even better

Q2.1: This is a multiline ode to
Cheese. I do enjoy it.

One time I bought too much Brie.

I tripped.

And I slipped.

And the cheese fell into the sea.

Q2.2: Did I mention Paneer? It's great too.

Q2.3: Blue Cheese is strong stuff

Q2.4: Is cheese used in staged delivery?

Q3.1: Up, Up, Down, Down, Left, Right, Left, Right, B, A