

1) You are *not* allowed use Eclipse for any part of this exam

2) There are 5 allowed Resources (and nothing more):

(1) OO Design Principles Handout

[https://rhit-](https://rhit-csse.github.io/csse220/Docs/Handouts/Basic_OO_Principles_for_CSSE220.pdf)

[csse.github.io/csse220/Docs/Handouts/Basic_OO_Principles_for_CSSE220.pdf](https://rhit-csse.github.io/csse220/Docs/Handouts/Basic_OO_Principles_for_CSSE220.pdf)

(2) UML Cheatsheet

https://rhit-csse.github.io/csse220/Docs/Handouts/UML_Cheatsheet.pdf

(3) Box & Pointer Cheatsheet

[https://rhit-](https://rhit-csse.github.io/csse220/Docs/Handouts/drawing_box_and_pointer.pdf)

[csse.github.io/csse220/Docs/Handouts/drawing_box_and_pointer.pdf](https://rhit-csse.github.io/csse220/Docs/Handouts/drawing_box_and_pointer.pdf)

(4) A single double-sided 8.5 x 11" paper cheatsheet that you made for yourself

(5) The Java API

<https://docs.oracle.com/javase/8/docs/api/>

3) Please keep phones **off the table** and turn off all chat functionality on your computer.

4) Diagrams

- Some questions below require you to draw a Boxes-and-pointers diagram
- You will be provided with paper for these questions. Please write your name at the top of every page and turn them in when you complete the exam.

5) In several places below, you will be asked to provide a UML diagram (or a recursive trace). The best option to give us a UML diagram below is to make it on plantUML using the link plantuml.com/plantuml and then saving the resulting png and uploading the png file for your submission to the question.

If you strongly desire to draw the UML diagram by hand, you may draw it on paper and convert to a PDF using your phone. If this is what you are doing, please inform your instructor before using your phone.

Instructions for converting a file to PDF:

To upload a scanned sheet of paper from your phone and generate a PDF:

- Adobe Scan is a free document scan app that is available for both Android and IOS devices.
- Android: https://play.google.com/store/apps/details?id=com.adobe.scan.android&hl=en_US
- IOS: <https://apps.apple.com/us/app/adobe-scan-digital-pdf-scanner/id1199564834>

There are also options like the built in Google Drive app. You can hit the + icon and click Scan.

- Android: https://play.google.com/store/apps/details?id=com.google.android.apps.docs&hl=en_US
- IOS: <https://apps.apple.com/us/app/google-drive/id507874739>

Other alternatives for scanning can be found on page 1 and 2 here:

- https://gradescope-static-assets.s3.amazonaws.com/help/submitting_hw_guide.pdf

HINT: If you have a file on your machine that is not in PDF form, you can usually opt to print the document and you should be able to save to a PDF from there.

In Windows, you must have Adobe installed (<https://helpx.adobe.com/acrobat/using/print-to-pdf.html>):

- Open a file in a Windows application
- Choose File > Print

- Choose Adobe PDF as the printer in the Print dialog box
- Click Print. Type a name for your file, and click Save

In Mac OS:

- Open the file in Preview
- Select File > Export to PDF
- Choose the location and name to save the PDF

Q2 Code output

8 Points

For each of the pieces of code below, please provide the resulting output.

Q2.1 Part 1

2 Points

```
3 public class ProgramOutput {  
4  
5     public static void main(String[] args) {  
6         int a = 2;  
7         int b = 11;  
8         double c = 11.0;  
9         System.out.println(b/a);  
10        System.out.println(c/a);  
11    }  
12  
13 }
```

Please type the output below (there should be 2 lines total of output for this problem):

5
5.5

Q2.2 Part 2

2 Points

```
1 public class ProgramOutput {  
2     public static void main(String[] args) {  
3         String output = "Exams make me feel happy!";  
4         String other = output.substring(6);  
5         output.replace("happy", "sad");  
6         String result = "Everything else does " + other;  
7         System.out.println(output);  
8         System.out.println(other);  
9         System.out.println(result);  
10        output = output.replace("happy", "sad");  
11        System.out.println(output);  
12        output = result;  
13        other = "Everything else does make me feel happy!";  
14        //in the following, boolean statements print "true" or "false"  
15        System.out.println("output == result : " + (output==result));  
16        System.out.println("result == other : " + (result==other));  
17        System.out.println("output.equals(other) : " + (output.equals(other)));  
18    }  
19 }  
20 }
```

Please type the output below (there should be 7 lines total of output for this problem):

Exams make me feel happy!
make me feel happy!
Everything else does make me feel happy!
Exams make me feel sad!
output == result: true
result == other: false
Q2.3 Part 3 output.equals(other): true

2 Points

```
3 import java.util.ArrayList;  
4  
5 public class ProgramOutput {  
6     public static void main(String[] args) {  
7         ArrayList<Integer> allNums = new ArrayList<Integer>();  
8         for(int i = 0; i <= 55; i+=5) {  
9             allNums.add(i);  
10        }
```

```

11 }
12
13     ArrayList<Integer> some = new ArrayList<Integer>();
14     ArrayList<Integer> others = new ArrayList<Integer>();
15     for(Integer val : allNums) {
16         if(val%2 == 0) {
17             some.add(val);
18         } else {
19             others.add(val);
20         }
21     }
22
23     //When printing ArrayLists, if the list contained values 1, 2 and 3
24     //the string printed would be: [1, 2, 3]
25     System.out.println("allNums is: " + allNums);
26     System.out.println("some is: " + some);
27     System.out.println("others is: " + others);
28
29     ArrayList<Integer> last = new ArrayList<Integer>();
30     for(int i = 0; i < some.size(); i++) {
31         last.add(some.get(i));
32         last.add(others.get(i));
33     }
34
35     System.out.println("last is: " + last);
36
37     System.out.println("last==allNums: " + (last==allNums)); //true or false
38     System.out.println("last.equals(allNums): " + last.equals(allNums));
39 }
40 }
```

Please type the output below (there should be 6 lines total of output for this problem):

allNums is: [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55]
 some is: [0, 10, 20, 30, 40, 50]
 others is: [5, 15, 25, 35, 45, 55]
 last is: [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55]
 last == allNums: false
 last.equals(allNums): true

Q2.4 Part 4

2 Points

```

3 import java.util.HashMap;
4
5 public class ProgramOutput {
6
7     public static void main(String[] args) {
8         double[][] doubs = new double[3][3];
9
10        for(int r = 0; r < doubs.length; r++) {
11            for(int c = 0; c < doubs[0].length; c++) {
12                doubs[r][c] = 0.5 + Math.pow(2, r*c);
13            }
14        }
15
16        HashMap<Integer, Double> map = new HashMap<Integer, Double>();
```

```
17     for(int i = 0; i < doubs.length; i++) {           '
18         map.put(i, doubs[i][i]);                         '
19     }                                                 '
20                                                 '
21     //HashMaps can iterate in random order, but you may simply '
22     //print the keys in order in your output (0, 1, 2,...) '
23     for(int key : map.keySet()) { '
24         System.out.println(key + " maps to " + map.get(key)); '
25     } '
26                                                 '
27     map.put(1, 5.75); '
28     System.out.println("1 maps to " + map.get(1)); '
29 } '
30 }
```

Please type the output below (there should be 4 lines total of output for this problem):

0 maps to 1.5
1 maps to 2.5
2 maps to 16.5
1 maps to 5.75

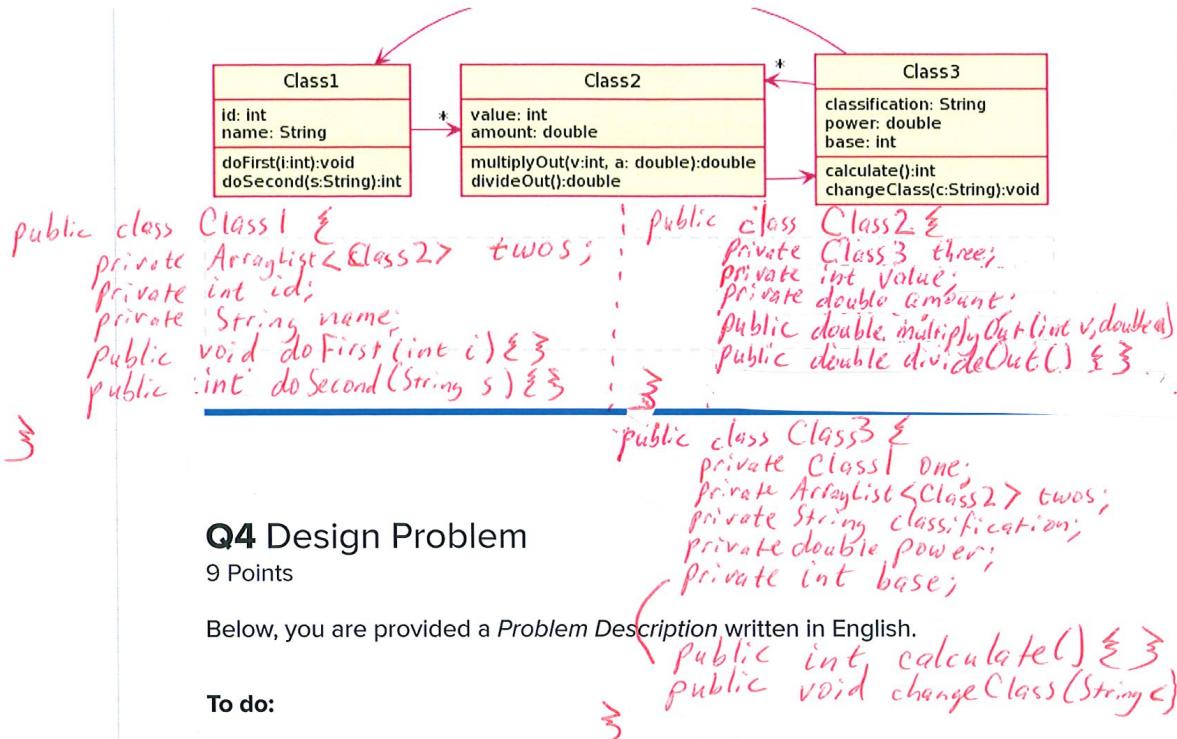
Q3 Convert UML to Code

4 Points

To Do:

- In the blank below, please type out the Java code that corresponds to the given UML diagram
- Include only the class declarations, all names and types of class variables and method declarations
- Leave the method bodies empty (do not provide code/implementation for any methods)

Reminder: In UML, when you see a ':' after a variable name, the word following the colon is the type of that variable. Similarly, when you see a ':' at the end of a method, the type there is the return type of the method.



Q4 Design Problem

9 Points

Below, you are provided a *Problem Description* written in English.

To do:

- Question 5.1: Solution A - please describe what is incorrect about this solution by giving first the principle(s) violated by number (e.g., 1a, 2b), and second a brief summary explanation for your first answer
- Question 5.2: Solution B - please describe what is incorrect about this solution by giving first the principle(s) violated by number (e.g., 1a, 2b), and second a brief summary explanation for your first answer
- Question 5.3: Please provide your design/solution for the *problem description*

THE PROBLEM DESCRIPTION

The CSSE department would like us to create an app that tracks all the students that a particular professor has had in a class. A professor has had a student in his/her class if the student has signed up for any one of his/her classes, even if the student later dropped the class. In this program, we only need to track the names and email addresses of both the professors and the students (assume that email addresses will be unique for all professors and

students), no other information is required for this application. The app must be able to do the following items:

- Add (and track) professors and students
- Add the case when a professor has had a student in a class
- Given a professor's email address, print all student names and email addresses who that professor has had in a class
- Given a student's email address, print all names and email addresses of the professors who have had this student in a class
- Given two professor email addresses, print all the students that both professors had in any class (the intersection of the sets of students)
- Given a student email address and a professor email address, print a statement indicating that the professor has had that student in at least one class, or never had that student in a class

Q4.1 Solution A

3 Points

Below is Solution A to the problem statement given above (repeated below for your convenience). This design is faulty in at least one major way, but there could be a number of issues. In the blanks below, you will provide the set of OO Design Principles that you believe Solution A violates. You will then be asked to explain why this principle is violated and/or why the design violates this principle.

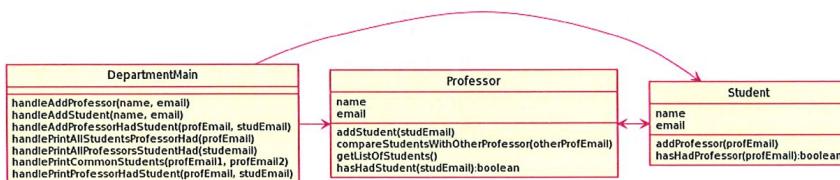
THE PROBLEM DESCRIPTION

The CSSE department would like us to create an app that tracks all the students that a particular professor has had in a class. A professor has had a student in his/her class if the student has signed up for any one of his/her classes, even if the student later dropped the class. In this program, we only need to track the names and email addresses of both the professors and the students (assume that email addresses will be unique for all professors and students), no other information is required for this application. The app must be able to do the following items:

- Add (and track) professors and students

- Add (and track) professors and students
- Add the case when a professor has had a student in a class
- Given a professor's email address, print all student names and email addresses who that professor has had in a class
- Given a student's email address, print all names and email addresses of the professors who have had this student in a class
- Given two professor email addresses, print all the students that both professors had in any class (the intersection of the sets of students)
- Given a student email address and a professor email address, print a statement indicating that the professor has had that student in at least one class, or never had that student in a class

Solution A:



In this blank, please identify the number/letter combination of the OO Design Principle(s) that are violated (i.e. 1.b, 2.a) by Solution A (you ONLY need to worry about the principles we have discussed in class thus far, no others):

1a

In this blank, please explain why or how the principle (or principles) is/are violated (i.e., what is wrong with Solution A?).

Main only stores 1 professor and 1 student
 Students track only 1 professor
 Professors only track 1 student

Q4.2 Solution B

3 Points

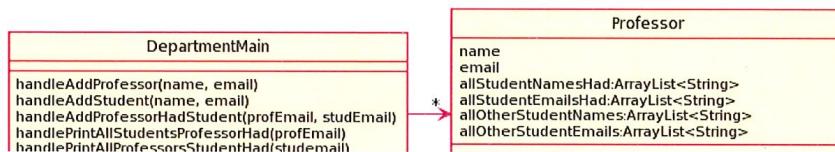
Below is Solution B to the problem statement given above (repeated below for your convenience). This design is faulty in at least one major way, but there could be a number of issues. In the blanks below, you will provide the set of OO Design Principles that you believe Solution B violates. You will then be asked to explain why this principle is violated and/or why the design violates this principle.

THE PROBLEM DESCRIPTION

The CSSE department would like us to create an app that tracks all the students that a particular professor has had in a class. A professor has had a student in his/her class if the student has signed up for any one of his/her classes, even if the student later dropped the class. In this program, we only need to track the names and email addresses of both the professors and the students (assume that email addresses will be unique for all professors and students), no other information is required for this application. The app must be able to do the following items:

- Add (and track) professors and students
- Add the case when a professor has had a student in a class
- Given a professor's email address, print all student names and email addresses who that professor has had in a class
- Given a student's email address, print all names and email addresses of the professors who have had this student in a class
- Given two professor email addresses, print all the students that both professors had in any class (the intersection of the sets of students)
- Given a student email address and a professor email address, print a statement indicating that the professor has had that student in at least one class, or never had that student in a class

Solution B:



handlePrintCommonStudents(profEmail1, profEmail2)	addStudent(studEmail)
handlePrintProfessorHadStudent(profEmail, studEmail)	compareStudentsWithOtherProfessor(otherProfEmail)
	getListOfStudents()
	hasHadStudent(studEmail):boolean

In this blank, please identify the number/letter combination of the OO Design Principle(s) that are violated (i.e. 1.b, 2.a) by Solution B (you ONLY need to worry about the principles we have discussed in class thus far, no others):

1c, 2a

In this blank, please explain why or how the principle (or principles) is/are violated (i.e., what is wrong with Solution B?).

1c - Student names and emails are duplicated for every professor.

2a - Student is a primary noun and should be a class.

Q4.3 Your Solution

3 Points

To Do:

- Please provide a UML diagram (from plantuml preferably) that presents a design that solves the *Problem Description* given above and does not violate any of the principles we have discussed in class so far
- You may choose to take Solution A or B and make improvements until the design no longer violates the principles originally violated by either Solution A or B, OR you may design something from scratch
- Your design must be able to accommodate each of the requirements stated above and repeated below for your convenience

THE PROBLEM DESCRIPTION

The CSSE department would like us to create an app that tracks all the students that a particular professor has had in a class. A professor has had a student in his/her class if the student has signed up for any one of his/her

classes. Note that a student later dropped the class. In this assignment this only

classes, even if the student later dropped the class. In this program, we only need to track the names and email addresses of both the professors and the students (assume that email addresses will be unique for all professors and students), no other information is required for this application. The app must be able to do the following items:

- Add (and track) professors and students
- Add the case when a professor has had a student in a class
- Given a professor's email address, print all student names and email addresses who that professor has had in a class
- Given a student's email address, print all names and email addresses of the professors who have had this student in a class
- Given two professor email addresses, print all the students that both professors had in any class (the intersection of the sets of students)
- Given a student email address and a professor email address, print a statement indicating that the professor has had that student in at least one class, or never had that student in a class

Here is the plantUML link to SolutionA:

http://www.plantuml.com/plantuml/uml/ZP8_ImGn4CNxV8gK2cUXfOZSeGSMoXrOMxSsumVuxCR27YTpUHR2QUAMoMtRptUvbvx9eO8i_EAdet_WWHd2R-i9YUqGm_YXei4EbxBj38xzFm4nkjFDO4VBMx6SH_30Ob2FFFWSANHWR7deZ_pF2C

Here is the plantUML link to SolutionB:

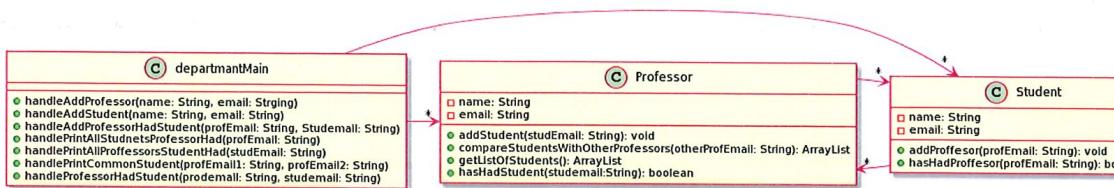
http://www.plantuml.com/plantuml/uml/ZP8zJyCm48Pt_mgh4w0mm5YXgX5KOW1QAGn_7fLzHDHQI2IJnZ6RD3oAoWhlqG3jjPOrlNEkmCIENzbmM0XqO3lrc-vwuEiGIJb7w6HBRXR1EvGb8usamZEwyibreFB4tpENXE9uh6NCJdon-TCSwENEwyAsJwV_yhyVyTIWURAH54HY-Yro8WxVKM3kCxBalluVIYY1_sx9KzhYPGNtgPsdA3Vjht9vORPxq7QVMAcTQy3YwUBi-jskvE9caJMADTgwC_e7

Below, you are asked to submit a file that contains your UML diagram. Please use plantuml ([plantuml.com/plantuml](http://www.plantuml.com/plantuml)) and save the image to a png and upload that file. *If you are taking this exam in another place outside the classroom, you may draw the UML diagram by hand and submit a scanned PDF (see instructions above), or you may also use PlantUML.*

 No files uploaded

No files uploaded

In the blank below, please provide any explanations in your design above that may help us understand what you were going for. This is optional, you may leave this field blank if you feel the design speaks for itself. If there is something you were trying to do, but couldn't quite get what you wanted, you can explain (VERY briefly) what it is you wanted to do that you couldn't:



Q5 True or False Questions

5 Points

For each of the questions below, please select either True or False

Q5.1

0.5 Points

If you had a class named **SomeClass** that had a **static** method named **someMethod()**, to call that method, you would say "**SomeClass.someMethod()**" because you do NOT need an instance of the **SomeClass** class to call the method.

True

False

Q5.2

0.5 Points

The **add** method of the **ArrayList** class is **static** because it can take either 1 or

2 arguments

True

False

Q5.3

0.5 Points

The following code will result with "Empty String!" being printed to the console

```
String str = "";
if(str == null) {
    System.out.println("Empty String!");
}
```

True

False

Q5.4

0.5 Points

The keyword "this" can and should be used for static variables/fields of a class
(as in this.var)

True

False

Q5.5

0.5 Points

The following returns either true or false, which will it return?

```
ArrayList<Integer> list1 = new ArrayList<Integer>();
ArrayList<Integer> list2 = list1;
list1.add(5);
list2.add(list1.get(0));
return list1 == list2;
```

True

False

Q5.6

0.5 Points

The following returns either true or false, which will it return?

```
String str1 = new String("Hello");
String str2 = new String("Hello There!");
return str2.substring(0,5).equals(str1);
```

True

False

Q5.7

0.5 Points

Constructors are special methods of a class that are used to initialize all the fields of the class and perform any other initializations needed.

True

False

Q5.8

0.5 Points

The code below will NOT call the constructor of the SomeClass class because it does not take any arguments

```
SomeClass sc = new SomeClass();
```

 True False**Q5.9**

0.5 Points

A class may have many constructors as long as they each take different parameter types in at least varying orders

 True False**Q5.10**

0.5 Points

In the following code, the value 0.5 will be printed to the console:

```
int a = 1;  
int b = 2;
```

```
System.out.println(a/b);
```

True

False