

Student Name: \_\_\_\_\_ Student ID # \_\_\_\_\_

**UOSA Statement of Academic Integrity**

*On my honor I affirm that I have neither given nor received inappropriate aid in the completion of this exercise.*

Signature: \_\_\_\_\_ Date: \_\_\_\_\_

**Notes Regarding this Examination**

**Open Book(s)** You may consult any printed textbooks in your immediate possession during the course of this examination.

**Open Notes** You may consult any printed notes in your immediate possession during the course of this examination.

**No Electronic Devices Permitted** You may not use any electronic devices during the course of this examination, including but not limited to calculators, computers, and cellular phones. All electronic devices in the student's possession must be turned off and placed out of sight (for example, in the student's own pocket or backpack) for the duration of the examination.

**Violations** Copying another's work, or possession of electronic computing or communication devices in the testing area, is cheating and grounds for penalties in accordance with school policies.

**Question 1:** Recursion and Iteration (35 points)

Recall that ‘%’ is the symbol for modulus arithmetic. That is, given two variables  $x$  and  $y$ , both of type `int`, the statement ‘ $x \% y$ ’ gives the remainder left over after dividing  $x$  by  $y$ . For example,  $5 \% 3$  is 2 because 3 goes into 5 once with a remainder of 2 left over. Similarly,  $3 \% 5$  is 3 because 5 goes into 3 zero times with a remainder of 3 left over.

Examine the code for `RecursiveRemainder.java`:

```
public class recursiveRemainder {

    public static int method1(int a, int b) {
        if (0 == b) return a;
        else return method1(b, a % b);
    }

    public static void method2(int first, int second) {
        for (int i = 1; i < first; i++)
            for (int j = 1; j < second; j++)
                System.out.println("method1 of " + i +
                    " and " + j + " is " + method1(i, j));
    }

    public static void main(String[] args) {
        method2(3, 3);
    }
}
```

A. Given the code above, what will be printed when this program is run? *Explain* your answer.

B. Given the code above, which methods are iterative, recursive, both, or neither? *Explain* your answer.

C. Given the code above, are any of the methods self-recursive, mutually recursive, both, or neither? *Explain* your answer.

D. In principle, should it be easy to convert the iterative parts of this code to recursive code that gives the same output given the same initial input? *Explain* your answer.

E. If you were to convert the iterative parts of this code to use recursion instead, would you expect it to run faster, slower, or exactly the same speed as the version given above? *Explain* your answer.

F. If you were to convert the iterative parts of this code to use recursion instead, would you expect it to use more, fewer, or exactly the same number of local variables as the version given above? *Explain* your answer.

**Question 2:** Preconditions, Recursion, and Iteration (10 points)

A. Given the code from Question 1, do you think that `method1` should have a precondition added to it? *Explain* your answer.

B. Given the code from Question 1, do you think that `method2` should have a precondition added to it? *Explain* your answer.

**Question 3:** Postconditions, Assertions, Exceptions, and OO Programming (10 points)

For each of the following method signatures, *explain* whether checking a postcondition in that method should be done with an assertion, a conditional with an exception, either, both, or neither.

A. `public Object methodA (Object o)`

B. `private Object methodB (Object o)`

**Question 4:** Preconditions and Exceptions (10 points)

Consider the following two approaches to dealing with an input parameter that is required to be non-negative:

Approach one:

```
private int exampleOne (int n) {
    if (n < 0) {
        System.out.println ("Input parameter cannot be negative.");
        return -1;
    }
    ...
}
```

Approach two:

```
private int exampleTwo (int n) throws IllegalArgumentException {
    if (n < 0) {
        throw (new IllegalArgumentException
            ("Input parameter cannot be negative.));
    }
    ...
}
```

Which approach should be considered preferable for checking this precondition? *Explain* your answer.

**Question 5:** Exceptions and Inheritance (10 points)

*Explain* the relationship between inheritance and the ordering of `catch` blocks associated with the same `try` block.

**Question 6:** Ethics (25 points)

Lisa works for OCRA, which provides web-based optical character recognition (OCR) services that allow people to upload images containing printed words and download the recognized text. OCRA targets ads to users based on the content of that text, in accordance with OCRA's terms of use as published on their website. A standard CAPTCHA ensures that each upload-download pair is carried out by a person who might see the ads, rather than by computers running scripts.

Jieping works for BuddySystem, a social networking site. They, too, are supported by ads, which are targeted based on what users say in their posts, in accordance with BuddySystem's own terms of use on its website.

Lisa and Jieping are best friends. One day while having breakfast together at a local restaurant they begin to muse about working together which leads them to speculate about who would switch companies to make it possible. Then it hits Lisa, lots of the photos that BuddySystem users upload contain text—OCRA could OCR those! "I'll bet a lot of your users would love that service."

"I'll bet we could learn a lot about our users from that text," Jieping notes.

They instantly agree that a quick and dirty prototype would be just the thing to convince their respective companies to form a partnership and before breakfast is over they have a rough design for the system and a plan for how to implement it. Jieping will write code to pull pics from BuddySystem and send them to OCRA's servers. Lisa will write code to allow Jieping's code to bypass the CAPTCHA as well as the code to send the OCR'ed documents back to BuddySystem. Jieping also will write the code to pass the returned text into BuddySystem's parser just as if it was text typed by a user. Working feverishly, they have the whole thing functioning by the end of the day and head out to celebrate together.

A. Find at least one ethical principle from a professional code of ethics that is relevant to this scenario. List the principle, give its source, and *explain* why you think it is relevant.

B. Pick Lisa or Jieping and say whether you think that person abided by (that is, followed) the principle you listed and *explain* how you came to that conclusion.

C. Give one likely motivation for that person's action and *explain* how you concluded that was a likely motivation.

D. List one ethical-decision-making problem (interfering factor) that is likely to have contributed to at least one of that person's decisions and *explain* how you concluded that was a likely problem.

E. List one ethical-decision-making strategy that person could employ to improve her or his ethical decision making and *explain* how she or he might employ that strategy in this situation.