Program Regrade Request
Attach to front of graded hardcopy
Be complete but concise

Name: ____________________________________________ Date: ________________

Assignment: ___________________________________ Class: ___________________

1. _____ (Initial to confirm) I understand that I may only request regrading when I have a credible justification. This is not an opportunity to request a regrade of an assignment just to see if I might get luck. I understand that abuse of the regrade request may result in a lower grade.

2. (Fill in the blank – e.g., love, despise) I _________________ fluffy bunnies and myself as a person.

3. _____ (initial to confirm) I have carefully read the test document. I have a copy of my final submission for the assignment, and I have executed the full test on my final submission. I conducted my tests on the department Linux boxes (unless otherwise specified) using any compile/JVM options given in the test document. Below are all additional special options, etc. that I used to execute the tests.

4. Clearly list the specific (sub)tests to be regraded in the specified format (see below). Failure to follow the format specification will result in a rejection. It is unlikely a rejection will be reversed.
Please only print and submit the previous page. I already have a copy of the following pages.

You may only submit a regrade request when such reconsideration is well justified. Do not abuse this mechanism.

There are two grounds for regrading a (sub)test:

**Correct Solution**
You believe your submission to be correct. You must include a justification(s) for this claim. Justifications include 1) the grading is simply incorrect according to the test, 2) the test does not match the specification, and 3) the specification is ambiguous and I interpreted it as follows, which differs from the test’s assumptions. Be sure to include your supporting evidence for each claim.

**Required format:**
<(Sub) Test(s)>:
Correct Solution
Justification: <Justification>

Example:
Test 2: Correct Solution
Justification: The specification states that the “name must be alphanumeric.” The test uses the name “Bob Smith”, which contains a space which is not alphanumeric.

**Reconsideration**
At the time of submission, you believed your solution to be correct and well tested; however, the test document focused on some particular element that you missed (e.g., your program works everyday but Wednesday). The burden is on **you** to demonstrate **all** of the following:

1. You heavily tested your code. The TA will presume your code failed because you didn’t test. Your required burden is to show the TA that this is not the case. The best response is “You can see my test script named X in my submission which covers a multitude of cases.” If you don’t have this, a possible response is “Here’s how I tested...” The TA should be able to replicate these passing tests on your final submission.
2. Some part of the test (which you missed) **disproportionately** impacted your grade.

A disproportionate impact is one where the flaw causes test failures unrelated to the actual flaw. Let’s consider some examples:

**No grounds for reconsideration:** Say your constructor disallows negative numbers even though you now realize such numbers are allowed by the specification. Tests focused on your constructor **should** fail. Such tests are **not** a disproportionate impact so you should **not** request regrading for such tests.

**Potential grounds for reconsideration:** Say you then have tests of an encode method that first call the constructor using a negative number. These tests may be focused on something other than the constructor and the tests happen to use negative numbers. Here you are failing tests that may not be directly related to the flaw. This may create a disproportionate impact on your grade. Of course, if the point of the test is to encode negative numbers then there’s no grounds for a regrade request.

**Absolutely no grounds for reconsideration:** Say the fix is really, really small. Do **not** submit a regrade request. It doesn’t matter that the changes to fix the problem are small.

For reconsideration, you **must** include all changes to your code required to fix the problem. These changes must be 1) specific and 2) simple. Specific changes give clear and concise instructions on what to change and where, not long paragraphs. Simple means the changes can be described in just a few lines. For example,

```
Change line 83 in Foo.java from
if (number <= 0)
to
if (number < 0)
```

Put yourself in the place of the TA. Could you quickly make and evaluate the changes given in your request if you were the TA?

Do not submit your revised code. The TA will only work from your final submission. If what you submitted isn’t easy to correct, then the change is not simple, in which case reconsideration should not be requested.
**Required format:**

<Sub>Test(s): Reconsideration

Testing: <Proof of Testing with explanation of how your tests missed this case>

Justification: <Justification of disproportionate impact>

Correction: <Specification of minor code correction so tests run>

Example:

Test 2: Reconsideration

Testing: In Test.java (included in my submission), you will see that I extensively tested the constructor. Unfortunately, I misread the specification and didn't realize the constructor could accept negative numbers for the ID. Note that I test the constructor with negative IDs but incorrectly expect an exception.

Justification: Test 2 focuses on encode(), but the object used for encoding is always constructed with a negative ID, which throws an exception before encode() is even called. While I understand encoding negative IDs is part of the test, this flaw causes me to fail all encoding tests.

Correction: Delete the if statement starting on line 83 in Person.java. This statement incorrectly throws an exception for negative IDs.