

# Java Basics Review: Problem Statement

## Background

In the Java Basics tutorial, we created a grading program which takes completed tests and returns a letter grade. These tests were basic in the sense that they didn't contain any questions. Instead, they just contained two values: the total number of questions and the number of correct questions.

## Problem Statement

Using the existing code, create a test grading system that supports real tests. The grader will read in a directory of completed tests then grade those tests based on a real rubric.

To simplify the problem a bit, here are a few assumptions:

- Assume that each question is multiple choice
- Assume that each test is in a text file in some directory
- Assume that the rubric is in a text file in some directory

## Requirements

The program must be called using the command line. In other words, the tool will need a main method. The tool must then be run in the following format:

```
java -jar Grader.jar /absolute/path/to/rubric.txt /absolute/path/to/test_1.txt ... [/absolute/path/to/test_n.txt]
```

The brackets are used to denote optional parameters. In this case, the program must accept at least on rubric and at least one test. Any additional tests are optional.

Each test will simply list the multiple choice answer for a given rubric on each line. For example, a test with eight questions might have a completed test of the following form:

```
A
B
A
D
C
D
A
A
```

The corresponding rubric will then list the number of questions on the first line, the grading breakdown on the following line, and each question on the remaining lines. For example, a rubric for the previous test should look like the following:

```
8
90;75;60;50
What color is the sky?;Blue:1;Red:0;Green:0;Purple:0
What color is grass?;Green:1;Blue:0;Red:0;Orange:0
...
...
...
...
```

Where is Tokyo?;Japan:1;China:0;Hawaii:0;Texas:0  
 Who drives the 48 car is NASCAR?;J. Johnson:1;J. Bieber:0;A. Ketchum:0;Me:0

The program must be able to parse both of these types of files and use them in a grading program. The actual implementation of the grading program is entirely up to the designer.

The program must output the results of each exam in the following format:

```
/absolute/path/to/test_1.txt letter_grade_1
...
/absolute/path/to/test_n.txt letter_grade_n
```

Upon completion, the program should be packaged as a jar and tested in various directories to ensure portability.

## Suggestions

This section offers up some suggestions for the implementation. It is advised that you take the time to learn the problem and draft up a design before reading this section. Naturally, there are several ways to produce a solution to this problem, so give it a shot first. By no means are the following suggestions the 'right' way of solving this problem. They simply offer some guidance for those that are stuck.

The first piece of advice is to model this problem using Java objects. Start by thinking about test features. For instance, a test has questions and answers. That gives us three quick classes right there: tests, questions, and answers.

A test class would have a list of questions while a question would have a list of answers. This structure alone should get you well on your way to developing a solution. Perhaps with these classes mapped out, you'll be able to see where the pieces fit together.

Maybe you'll want a rubric class, or maybe that doesn't make sense. After all, a rubric is just a completed test. It's totally up to you! If you get stuck, feel free to browse the solutions. Good luck!