CSE142—Computer Programming I
Programming Assignment #4
due: Tuesday, 4/26/22, 11 pm

This assignment will give you practice with interactive programs, if/else statements and methods that return values. Your program will prompt the user for information about two applicants and compute an overall score for each applicant. This is a simplified version of a program that might be used for admissions purposes.

The sample log of execution indicates how your program is to behave. For each applicant, we prompt for exam scores (either SAT or ACT) and overall GPA. The exam and GPA information is turned into numbers between 0 and 100 and these two scores are added together to get an overall score between 0 and 200. After obtaining scores for each applicant, the program reports how they compare.

Notice that the program asks for each applicant whether to enter SAT scores or ACT scores (SAT scores are integers that vary between 200 and 800, ACT scores are integers that vary between 1 and 36). In the case of SAT scores, the user is prompted for SAT math, reading, and writing scores. In the case of ACT scores, the user is prompted for English, math, reading and science scores. These scores are turned into a real-valued number between 0 and 100 using the following formulas:

$$\text{For SAT Scores: } \frac{2 \cdot math + reading + writing}{32}$$

$$\text{For ACT Scores: } \left\lceil \frac{English + 2 \cdot math + reading + science}{1.8} \right\rceil$$

After computing this exam score, we compute a number between 0 and 100 based on the GPA. The program prompts for the GPA, the maximum GPA, and a transcript multiplier. All three of these are positive real values (i.e., they can have a decimal part). The transcript multiplier is a value between 0.8 and 1.0 that the admissions staff use to account for differences across students and across schools. For example, a student who takes more AP courses or a student who comes from a high school that is known to have tough grading standards will get a higher transcript multiplier. You should turn this into a score between 0 and 100 using the following formula:

$$\frac{actual\ gpa}{max\ gpa} \cdot 100 \cdot transcript\ multiplier$$

At this point your program has two scores that vary from 0 to 100, one from test scores and one from GPA. The overall score for the applicant is computed as the sum of these two numbers (exam result + gpa result). Because each of these numbers is between 0 and 100, the overall score for an applicant ranges from 0 to 200.

As indicated in the sample log of execution, your program is to report the exam and GPA subscores and the overall score for each applicant. These should be rounded to one decimal place when displayed. You are required to do this using the round1 method described in lecture (see BMI2.java) or using the printf statement described in section 4.3 of the textbook (the alternative is to have redundant expressions throughout your code, which would be bad style). In addition to reporting the score for each applicant, the program should also produce whichever of the following messages is appropriate:

    The first applicant seems to be better
    The second applicant seems to be better
    The two applicants seem to be equal

You do not have to perform any error checking. We will assume that the user enters numbers that are in the appropriate range. Even though your program is rounding numbers to one decimal place when they are displayed, it should not round the numbers that are used to compute results and to compare scores. This can lead to some confusing output. For example, if the exam subscore is 84.06 and the GPA subscore is 62.78, then

they add up to 146.84. But rounding the displayed numbers to one digit, these would be displayed as 84.1 plus 62.8 adding up to 146.8. This is the correct output to produce even though it doesn't seem to add up properly.

You should use static methods to eliminate redundant code and to break the problem up into logical subtasks. Your main method should be short so that a person can easily see the overall structure of the program. You should avoid "chaining" long sequences of method calls together without returning to main. You are to introduce at least five static methods other than main and round1 to break this problem up into smaller subtasks and you should make sure that no single method is doing too much work. In this program, **none of your methods should have more than 15 lines of code in the body of the method** (not counting blank lines or lines with just curly braces on them). The 15-line limitation is a special requirement for this assignment because we want you to practice breaking up a program into methods. Be sure to once again include a short comment at the beginning of your program as well as a short comment for each method describing what it does. Look at the program BMI2.java discussed in lecture for a good example of eliminating redundant code and breaking a problem up into logical subtasks. See also the discussion of this program in section 4.5 of the textbook, especially the section on "Procedural Design Heuristics" starting on page 295 that explores issues like chaining. Also remember that because this program involves both integer data and real data, you need to use appropriate type declarations (type int and calls on nextInt for integer data, type double and calls on nextDouble for real-valued data). Finally, you should construct only one Scanner object for console input.

Name your program Admit.java and to access Scanner include the following at the beginning of your program:

```
import java.util.*;
```

This handout has a sample execution and the output comparison tool on the class web page has other samples. You are required to exactly reproduce the format of these sample executions. Limit yourself to constructs from chapters 1 through 4 of the textbook. You should not be using for loops as a way to solve this problem.

Sample log of execution (user input bold and underlined)

```
This program compares two applicants to
determine which one seems like the stronger
applicant.  For each candidate I will need
either SAT or ACT scores plus a weighted GPA.

Information for applicant #1:
    do you have 1) SAT scores or 2) ACT scores? 1
    SAT math? 450
    SAT critical reading? 530
    SAT writing? 490
    exam score = 60.0
    overall GPA? 3.4
    max GPA? 4.0
    Transcript Multiplier? 0.9
    GPA score = 76.5

Information for applicant #2:
    do you have 1) SAT scores or 2) ACT scores? 2
    ACT English? 25
    ACT math? 20
    ACT reading? 18
    ACT science? 15
    exam score = 54.4
    overall GPA? 3.3
    max GPA? 4.0
    Transcript Multiplier? 0.95
    GPA score = 78.4

First applicant overall score  = 136.5
Second applicant overall score = 132.8
The first applicant seems to be better
```