American University of Beirut
Department of Computer Science
CMPS 200 – Introduction to Programming
A04 – Deadline: Monday March 21 @11:00 PM

Exercise 1. Admission

Write a program Admission.java that might be used for admissions purposes. The program should prompt the user for information about two applicants and that computes an overall score for each applicant. For each applicant, the program prompts for exam scores (either SAT or ACT) and overall GPA. The exam information is turned into a number between 0 and 100 and the GPA information is turned into a number 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 which one looks better or whether they look equal.

The program should prompt for each applicant name then whether to enter SAT scores or ACT scores:

- SAT scores are integers that vary between 200 and 800. The user is prompted for SAT verbal and SAT math scores.
- ACT scores are integers that vary between 1 and 36. In that case, the user is prompted for English, Math, Reading and Science scores.

These scores are turned into a number between 0 and 100 using the following formulas:

For SAT Scores: $\frac{2\ x\ verbal+math}{24}$ and for ACT Scores: $\frac{2\ x\ reading+English+Math+Science}{1.8}$

These formulas produce numbers in the range of 0 to 100. After computing this exam score, we compute a number between 0 and 100 based on the GPA. Therefore the program prompts for the GPA and the maximum GPA. Both the GPA and maximum GPA are real values that should be turned into a score between 0 and 100 using the following formula:

$$\frac{actual\_gpa}{\max\_gpa}\ x\ 100$$

At this point your program has two scores that vary from 0 to 100, one from their test score and one from their 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.

You can assume that the user enters numbers that are in the appropriate range. 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 are to introduce at least five static methods other than main** to break this problem up into smaller subtasks and you should make sure that no single method is doing too much work.

You can find a sample run on the next page:

```
-----------------------------------------------------
This program will compare two candidates and determine
which one to be admitted to the department!
-----------------------------------------------------
Information for the first applicant:
        Full Name: Johnny Bran
        SAT (1) or ACT (2)?1
        SAT Math?: 430
        SAT Verbal?: 540
        Overall GPA? 3.7
        Maximum GPA? 4.0
Information for the second applicant:
        Full Name: Sarah Brown
        SAT (1) or ACT (2)?2
        ACT English?: 25
        ACT Math?: 32
        ACT Reading?: 22
        ACT Science?: 31
        Overall GPA? 3.8
        Maximum GPA? 4.0
Johnny Bran 's overall average = 155.4167
Sarah Brown 's overall average = 168.3333
It seems Sarah Brown is the best applicant
```

Use printf to format the output like this

Exercise 2. Guessing Numbers

Write an interactive program that prompts the user to think of an integer between one and a thousand and then asks the user to truthfully respond with true or false to statements of the form "My number is less than or equal to x?" for various values of x, until the program can deduce the chosen integer. Your program should use a strategy known as binary search for guessing. Binary search chooses the middle of the search interval as the value of x at every iteration.

Exercise 3. Credit Card

Say that you owe the credit card company `X` dollars. The company charges you an interest of `P` percent per month on the unpaid balance. You have decided to stop using the card and to pay off the debt by making a monthly payment of `N` dollars a month. Write a program, `CreditCard`, that writes out the balance and total payments made for every succeeding month until the balance is zero or less. For each month, calculate the interest due on the unpaid balance; then calculate the new balance by adding the interest and subtracting the payment. For instance, for a bill of $1000.0, interest rate of 1.5% and monthly payment of $100.0, the program's output will be:

```
The bill is: $1000.0
The monthly payment is: $100.0
The interest rate is: 1.5%

Month: 1  balance: 915.0               total payments: 100.0
Month: 2  balance: 828.725             total payments: 200.0
Month: 3  balance: 741.155875          total payments: 300.0
Month: 4  balance: 652.273213125       total payments: 400.0
Month: 5  balance: 562.057311321875    total payments: 500.0
Month: 6  balance: 470.4881709917031   total payments: 600.0
Month: 7  balance: 377.54549355657866  total payments: 700.0
Month: 8  balance: 283.20867595992735  total payments: 800.0
Month: 9  balance: 187.4568060993263   total payments: 900.0
Month: 10 balance: 90.26865819081618   total payments: 1000.0
Month: 11 balance: -8.377311936321576  total payments: 1100.0
```

Exercise 4. Monte Carlo PI Estimation

Monte Carlo simulations are techniques that consist of choosing sample experiments at random from a large set and then making deductions on the basis of the probabilities estimated from the result of these experiments. In this problem, you are to implement a Monte Carlo method for estimating the value of π.

Consider the first quadrant of a unit circle centered at (0,0). This quarter circle lies inside a unit square. A point with coordinates (x,y) is inside the quarter circle if $x^2 + y^2 <= 1$. The area of the quarter circle region can be estimated by picking, at random, points (x,y) that lie in the unit square, and for each point determining whether the point lies in the region. The fraction of points that fall in the region should give an estimate of π/4 (ratio of the area of the region and the area of the enclosing unit square). Multiplying by 4 gives an estimate of π.

Write a program `MonteCarloPi.java` that prompts the user to enter an integer n and prints an estimate of π using n random points as described above. How good are the estimates with N = 10, 100, 10000?