American University of Beirut
Department of Computer Science
CMPS 200 – Introduction to Programming
Assignment 2: Deadline Feb 28, 23:00

Exercise 1. Enough Time

Write a method, **enough_Time(h1, m1, h2, m2)**, that takes 4 integer parameters: hours_1, minutes_1, hours_2, minutes_2. These parameters represent two 24-hour times. For example, the two times 4:35 PM and 7:20 will be represented by the following parameters: (16, 35, 19, 20).

The function should return true if the gap between the two times is enough for a nap (greater than 40 minutes). It should return false otherwise. You may assume all parameters are valid, and both times represent times in the same day. If the second time is earlier than the first one, your function should return false. Test your function on different inputs from the command line.

`java EnoughTime 09:30 10:25 → true`

`java EnoughTime 18:20 17:30 → false`

`java EnoughTime 14:30 15:00 → false`

Exercise 2. Two equal One

Implement a method that takes three integers (num1, num2, num3) and returns *true* whenever the sum of any of these two integers is equal to the third integer. Note that numbers can be passed to the method in any order. Test your method by taking three numbers as command line arguments.

`java TwoEqualOne 20 30 50 → True`

`java TwoEqualOne 30 50 20 → True`

`java TwoEqualOne 10 5 45 → False`

Exercise 3. Normal Heart Rate

A normal heartbeat per minute for non-exercising person is between 60 and 100. Write the code that reads the heartbeat from the user, and whether he/she is exercising or not. It will then display either normal or abnormal.

Sample run:

```
Enter your heart beat rate: 75
Are you exercising? (yes/no): no
Normal heart beat rate.
```

```
Enter your heart beat rate: 120
Are you exercising? (yes/no): no
Abnormal heart beat rate.
```

```
Enter your heart beat rate: 125
Are you exercising? (yes/no): yes
Normal heart beat rate.
```

Exercise 4. Passed All or Failed Some

Write a program that reads three quizzes grades from the command line. It will then display either: "Passed all", when all grades are passing ones, or "Failed some" when there is at least one failing grade.

`Java quizzes 85.5 91 75.5 → Passed all`

`Java quizzes 85 51.5 75.5 → Failed some`

Exercise 5. Triangle Types

Write a program that asks the user to enter three sides of a triangle, and then displays its type. The possible types are: Scalene, Isosceles (two equal sides), and Equilateral (all three sides are equal). However, before checking the type, you have to validate that the three sides form a valid triangle: the sum of any two sides is greater than the third side.

Exercise 6. String Comparison

Write a method that takes three strings, and returns true if the strings are in alphabetical order (either ascending or descending), and false otherwise. Hint: you can use the string method ***compareTo***. To you're your work, implement the main method to take three words as input from the user.

Sample Runs:

```
Enter three words: Sun Moon Rain
They are not ordered.
```

```
Enter three words: Moon Rain Sun
They are ordered alphabetically
```