# CO322: Data Structure and Algorithms

## Lab2: Odd man out!

**Aim:** Aim of this laboratory is to develop simple algorithm(s) based on what we have already learnt in class and apply the algorithms analysis techniques to analyses them. Further, the lab class is aimed at enhancing your programming skills and your thinking abilities.

## Introduction:

The aim of this lab class is to find the most odd man! That is the man who is different from all others. For starters, if we have 3 numbers; say {1, 4, 3} then the odd man (or the number in this case) is 1. The way we find the odd number is by paring numbers based on their distance; the distance between 1 and 3 is 2 and the distance between 3 and 4 is just 1. So 3 and 4 are paired which leaves us with 1 unpaired making it the odd man!. Of cause we need to have odd number of entries for this to work.

When pairing there is no threshold; we pair the ones with the least distance between them. For example say the numbers we have are {100, 400, 300} then the odd number is 100. If we have similar distances then the result is unpredictable; that is  if our numbers are {1, 2, 3} then the odd man can be either 1 or 3 (but never 2 since it has to be paired with either 3 or 1).

There are 69 students in your batch so our aim is to find the odd man out of them. The odd man will get a prize which I will disclose later.

Each of you will be allowed to enter height, weight, gender and age together with your e-number. These values will be entered via a web-page to a MYSQL database. Your program should read these values by connecting to the MYSQL database. For convenience you will develop the program using Java.

Once all the values are read you need to *combine* them in some sensible way to have a single number representing the person. For simplicity we will just add all the numbers (height, weight and age) and call it the *personality* number. Females are usually small made than male, so if you are a female scale up the personality number 25% so that all will have a equal playing ground.

Next task is to pair up records based on the personality number. For every record you should find another record with the smallest difference in the personality number. Note that this is tricky than it might first appear; for example *X* might be closest to *Y*, but *Y* might be closer to some other *Z*. Once all records are paired the one which is left out will be the *odd man!* Your program should display the e-number of the odd man.

## Rules

You cannot use any $O(N^2)$ algorithms for this. In addition to your code you need to submit a report outlining your algorithms and analyzing their complexities. As usual you can submit *.txt or *.pdf.

**Other resources:**
We will provide a webpage to which you can enter the above details. Also we will provide some sample code showing how to connect to a MYSQL database and fetch data.


**Submission:** Please submit your Java source code file and the report to CMS links. Note that there are two links one for code and one for report. **Strictly no late submissions.**