# COMP108 Algorithmic Foundations
## Tutorial 5
### w/c 3rd March 2014

*Tutorial participation contributes to 5% of overall marks. For this tutorial, make sure you have scanned your ID card.*

1. Download two java files SortApp.java and Sort.java from the tutorial page
   `http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201314/tutorial.html`
   (Use right mouse click to save the files.)

   You can refer to the lecture notes for the pseudo codes. `http://www.csc.liv.ac.uk/~pwong/teaching/comp108/201314/notes.html` (Time complexity + Searching)

   (a) Compile and run the program; then enter some numbers, one per line, followed by $-1$ to terminate the input. Try the options to sort the numbers using different sorting methods. Note that these two functions are NOT working yet.

   (b) Fill in the program Sort.java the method **swap( )** to swap two entries `array[x]` and `array[y]`.

   (c) Fill in the program Sort.java the method **ssort( )** to sort the numbers in ascending order using the **selection sort** algorithm and test if it works.

   The array to be sorted is `data2[]`, which is a duplicate of the content of `data[]`. The variable `count` stores how many values the user has input. The size of the array `data2[]` has been set to `count`.

   You can make use of the **swap( )** method if necessary. *Remember to read the comments in the method.*

   Test cases:

   i. **10, 30, 20, 40, 50**
   ii. **50, 30, 10, 40, 20**
   iii. **50, 40, 30, 20, 10**
   iv. **40**, **20**, **60**, $-30$, $-40$, **10**, $-5$, $-50$

   (d) Fill in the program Sort.java the method **bsort( )** to sort the numbers in ascending order using the **bubble sort** algorithm and test if it works. The array to be sorted is `data2[]`, which is a duplicate of the content of `data[]`. You can make use of the **swap( )** method if necessary. *Remember to read the comments in the method.*

2. [**Puzzle**] An $8 \times 8$ chessboard has had two of its diagonally opposite squares removed, leaving it with sixty-two squares. It is said that we cannot tile the chessboard with 31 non-overlapping $2 \times 1$ rectangles (dominoes). Do you agree with it? Why?