Dr Tomasz Kurowski
t.j.kurowski@cranfield.ac.uk

# Practical Session 03
# Arrays

1. Write a program that creates a list of random numbers. Let the user specify the length of the list through an input dialog box

2. Write a method to sum all the integers in a matrix of integers. Use { {1, 2, 5}, {6, 7, 9}, {10, 11, 13}, {14, 15, 17}} to test it.

3. Design a method to create N random integer values in the range 0 to 100, and store them in an integer array of dimension N. Duplicate values are allowed. N can be entered by the user or set as a constant in your program.

4. Write a method to display all the elements of an integer array (of any size) to the console. Use it to test the method designed in step 2.

5. Design a method to sort an array of integers into ascending order, using your own implementation of the bubble sort algorithm. The bubble sort algorithm works by comparing each item in the list with the item after it, and swapping them if required ((if the first value is greater than the next one). The algorithm moves along the list, performing this operation on every element in turn. The algorithm then repeats this process again and again until it makes a pass all the way through the list without swapping any items (in other words, all items are in the correct order).

This technique implemented by this algorithm causes larger values to "bubble" to the end of the list while smaller values "sink" towards the beginning of the list.

A <u>single pass</u> through the algorithm will look something like this :

(This is 'pseudo code', designed to illustrate the algorithm, not to be syntactically complete)

```
swapped = false;                         // A Boolean
flagfor (j = 1; j <= array_size; j++) {

    if (numbers[j-1] > numbers[j]) {                    // Are 2 values out of order
                                                ?

        temp = numbers[j-1];                            // Yes .. swap them
        overnumbers[j-1] = numbers[j];

        numbers[j] = temp;

        swapped = true;                                 // And set flag accordingly

    }

}
```

6. Use the methods created in steps 4, 5 and 6 to create an array of random numbers, display them to the console, sort the array into ascending order, then display the array again to the console.