# CS19001: Programming and Data Structures Laboratory

Aritra Hazra;
CSE, IIT Kharagpur

http://cse.iitkgp.ac.in/~aritrah/course/lab/PDS/Autumn2019/

30-Aug-2019

# Arrays

Arrays are our first example of *structured data*.

## Declaration

```
int A[400];
float B[123];
```

Each of the elements A[0],A[1],...,A[399] is a variable of type int

## Declaration and Initialization

```
int A[5] = { 51, 29, 0, -34, 67 };
char C[4] = { ’g’, ’o’, ’d’, ’\0’ };
char C[4] =  "god";

printf("%d\n",A[5]); /*prints a garbage value*/
```

# Nested loops

One or more loops can be nested inside another loop.

## Sorting

Suppose you have an array A of n elements (say, integers). They are stored in the array locations

```
A[0], A[1], ..., A[n-1]
```

We want to rearrange these integers in such a way that after the rearrangement we have

```
A[0] <= A[1] <= A[2] <= ..... <= A[n-1]
```

The resultant array is **sorted**. There are many such sorting methods. One is bubble sort.

# Bubble sort

## Code

```
for (i=n-2; i>=0; --i)
{
  for (j=0; j<=i; ++j)
  {
    if (A[j] > A[j+1])
    {
      t = A[j];
      A[j] = A[j+1];
      A[j+1] = t;
    }
  }
}
```

## A[4]=4,3,2,1

i,j: A → $A'$

2,0: 4,3,2,1 → 3,4,2,1

2,1: 3,4,2,1 → 3,2,4,1

2,2: 3,2,4,1 → 3,2,1,4

bubble till position i=4-2=2.

1,0: 3,2,1,4 → 2,3,1,4

1,1: 2,3,1,4 → 2,1,3,4

bubble till position i=1

0,0: 2,1,3,4 → 1,2,3,4

bubble till position i=0

# Thank You