An algorithm that shuffle a set

You can use the Fisher-Yates shuffle. Here is the algorithm, listed on http://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle

```
To shuffle an array a of n elements (indices 0..n-1):
  for i from n − 1 downto 1 do
       j ← random integer with 0 ≤ j ≤ i
       exchange a[j] and a[i]
```

This can be solved by any computer language: C#, C, C++, Java, Python and so on. Note that n can be any size. The random method will be used to get a randome number j with 0 <= j <= i

Try to visualize the solution on paper. Let us say that you have an array of 5 elements: 1,2,3,4,and 5. Indices will be 0,1,2,3 and 4. There will be 4 iterations, per algorithm description.

First iteration

- Value of i: 4
- Use random function to get a value between 0 and 4. Let's say we got j=3, (a[3] is 4).
- Exchange a[4] and a[3], (that is to say the array now is 1,2,3,5 and 4)

Second iteration

- Value of i: 3
- Use random function to get a value between 0 and 3. Let's say we got j=0, (a[0] is 1).
- Exchange a[3] and a[0], (that is to say the array now is 5,2,3,1 and 4)

Third iteration

- Value of i: 2
- Use random function to get a value between 0 and 2. Let's say we got j=1, (a[1] is 2).
- Exchange a[2] and a[1], (that is to say the array now is 5,3,2,1 and 4)

Fourth iteration

- Value of i: 1
- Use random function to get a value between 0 and 1. Let's say we got j=0, (a[0] is 5).
- Exchange a[1] and a[0], (that is to say the array now is 3,5,2,1 and 4)

Final set: 3,5,2,1, and 4. At another run you may get the result set as "5 2 4 1 and 3"