Bubble Sort

ð The list is divided into two sublists: sorted and unsorted.

ð The smallest element is bubbled from the unsorted list and moved to the sorted sublist.

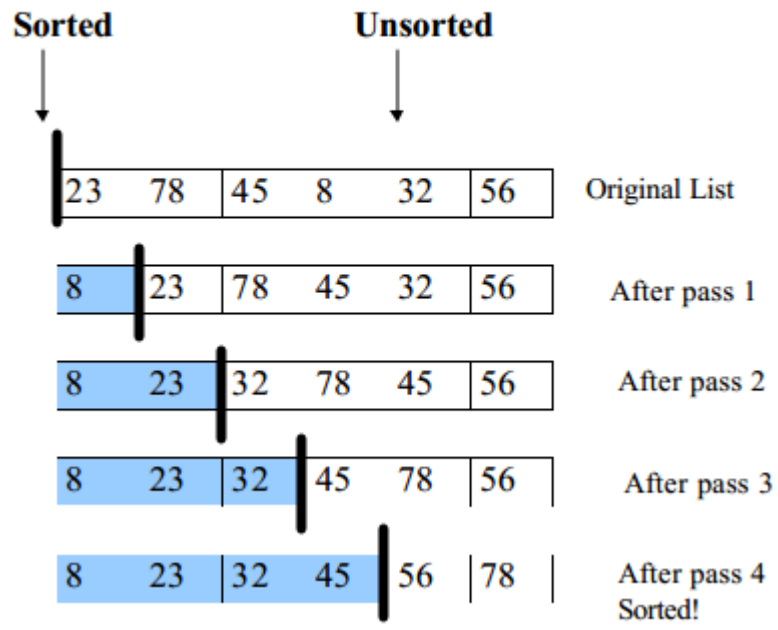
ð After that, the wall moves one element ahead, increasing the number of sorted elements and decreasing the number of unsorted ones.

ð Each time an element moves from the unsorted part to the sorted part one sort pass is completed.

ð Given a list of n elements, bubble sort requires up to n-1 passes to sort the data.

ð Bubble sort was originally written to “bubble up” the highest element in the list. From an efficiency point of view it makes no difference whether the high element is bubbled or the low element is bubbled.

Bubble Sort Example



CODES(C)

```c
#include <stdio.h>

int main()
{
  int array[100], n, c, d, swap;

  printf("Enter number of elements\n");
  scanf("%d", &n);

  printf("Enter %d integers\n", n);

  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);

  for (c = 0 ; c < ( n - 1 ); c++)
  {
    for (d = 0 ; d < n - c - 1; d++)
    {
      if (array[d] > array[d+1]) /* For decreasing order use < */
      {
        swap       = array[d];
        array[d]   = array[d+1];
        array[d+1] = swap;
      }
    }
  }

  printf("Sorted list in ascending order:\n");

  for ( c = 0 ; c < n ; c++ )
     printf("%d\n", array[c]);

  return 0;
}
```

CODES(JAVA)

```java
import java.util.Scanner;

class BubbleSort {
  public static void main(String []args) {
    int n, c, d, swap;
    Scanner in = new Scanner(System.in);

    System.out.println("Input number of integers to sort");
    n = in.nextInt();

    int array[] = new int[n];

    System.out.println("Enter " + n + " integers");

    for (c = 0; c < n; c++)
      array[c] = in.nextInt();

    for (c = 0; c < ( n - 1 ); c++) {
      for (d = 0; d < n - c - 1; d++) {
        if (array[d] > array[d+1]) /* For descending order use < */
        {
          swap       = array[d];
          array[d]   = array[d+1];
          array[d+1] = swap;
        }
      }
    }

    System.out.println("Sorted list of numbers");

    for (c = 0; c < n; c++)
      System.out.println(array[c]);
  }
}
```