



Department of IT and Computer Science
Pak-Austria Fachhochschule: Institute of Applied Sciences
and Technology, Haripur, Pakistan

COMP-201L Data Structures and Algorithms Lab

Lab Report: 04

Class: **Computer Science**
Name: **Yaseen Ejaz Ahmed**
Registration No.: **B20F0283CS014**
Semester: **Third**
Submission Date:
Submitted to: **Dr. Rafi Ullah**

Instructor Signature

Lab No. 4

Sorting Algorithms: Bubble Sort and Selection Sort

Objectives:

To understand & implement the working of sorting algorithms using arrays in C++

Tools/Software Required:

C++ Compiler

Introduction:

It is often necessary to arrange the elements in an array in numerical order from highest to lowest values i.e., from ascending to descending and vice versa. If an array contains string values or alphabetical order, then arrays need to be sorted. The process of sorting an array requires the exchanging of values. While this seems to be a simple process, a computer must be careful that no values are lost during this exchange.

Lab Tasks:

Lab Task 01: Write a code to sort the following arrays using selection sort method
[10, 34, 2, 56, 7, 67, 88, 42]

Code:

```
#include <iostream>
using namespace std;

BubbleSort(int a[],int n)
{
    int temp;

    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n-i;j++)
        {
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }

    cout<<"\n\nSorted in Ascending : \n";
    for(int i=0;i<n;i++)
        cout<<a[i]<<"\t";

}

int main()
{
    int size=8;

    int a[size]={ 10,34,2,56,7,67,88,42};

    cout<<"Original Array : \n";
```

```
        for(int i=0;i<size;i++)  
            cout<<a[i]<<"\t";  
  
        BubbleSort(a,size);  
    }  
}
```

Output:

```
Original Array :  
10      34      2      56      7      67      88      42  
  
Sorted in Ascending :  
2       7       10     34     42     56     67     88
```

Lab Task 02: Write a code to sort the following arrays using bubble sort method.
[10, 34, 2, 56, 7, 67, 88, 42]

Code:

```
#include <iostream>
using namespace std;
```

SelectionSort(int a[],int n)

```
{
    int temp,min;

    for(int i=0;i<n;i++)
    {
        min=i;

        for (int j = i+1; j < n; j++)
        {
            if (a[min]>a[j])
                min = j;
        }

        temp=a[i];
        a[i]=a[min];
        a[min]=temp;
    }

    cout<<"\n\nArray in Ascending order:\n";
    for(int i=0;i<n;i++)
        cout<<a[i]<<"\t";
}
```

int main()

```
{
    int size=8;
    int a[size]={ 10,34,2,56,7,67,88,42};

    cout<<"Original Array:\n";
    for(int i=0;i<size;i++)
        cout<<a[i]<<"\t";

    SelectionSort(a,size);
}
```

Output:

```
Original Array :  
10      34      2      56      7      67      88      42  
  
Sorted in Ascending :  
2       7       10      34      42      56      67      88
```

Results & Observations:

In this lab, we have learnt two different types of sorting an array. We do not use one type of sorting for all arrays as the conditions may differ. These types can be used for different reasons and different problems.