DSA
 Data Structures and Algorithms
 Problem Solving Skills

1. Sorting Alogrithm
      a. Selection Sort
      b. Buble Sort
      c. Quick Sort
      d. Merge Sort
      e. Heap Sort
        f. Insertion Sort

++++++++++++++
Selection Sort
++++++++++++++
```
ALGORITHM :: Selection_Sort(arr[0,....n-1],n)
//Sort the data in Ascending order using selection sort
//input  :: An array a[0,...n-1] of orderable elements
//output :: An array a[0....n-1] of ordered elements

for i<-0 to n-2 do
      min<-a[i]
      k<-i

  for j<-i+1 to n-1 do
       if a[j]<min then
            min <- a[j]
             k  <- j

  //exchange a[i] and a[k]
```

Code using Java
+++++++++++++++
```
import java.io.*;
import java.util.*;

//Client Code
public class Test
{
   public static void main(String[] args)throws IOException
      {

         Scanner scan = new Scanner(System.in);
         System.out.print("Enter the size of an Array :: ");
         int n = scan.nextInt();

          //Create an Array of size n
         int[] arr = new int[n];

         //Fill the array elements from the user
         for (int i = 0;i<=n-1;i++ )
         {
             System.out.print("Enter the array element :: ");
             arr[i] = scan.nextInt();
         }

         System.out.println("Array before Sorting :: "+Arrays.toString(arr));
```

```
         selectionSort(arr,n);

         System.out.println("Array before Sorting :: "+Arrays.toString(arr));

      }
      public static void selectionSort(int[] arr,int n){
            //logic of SelectionSort
            for (int i =0; i<=n-2; i++ )
            {
                  //setup used for sorting
                  int min = arr[i];
                  int k   = i;

                  for (int j = i+1;j<=n-1;j++ )
                  {
                        //Check for min element
                        if (arr[j] < min)
                        {
                              min = arr[j];
                              k   = j;
                        }
                  }

                  //Exchange arr[i] and arr[k]
                  int temp = arr[i];
                  arr[i] = arr[k];
                  arr[k] = temp;
            }
      }
}


++++++++++
Bubblesort
++++++++++
ALGORITHM :: Bubble_Sort(arr[0,....n-1],n)
//Sort the data in Ascending order using bubble sort
//input  :: An array a[0,...n-1] of orderable elements
//output :: An array a[0....n-1] of ordered elements

for i<-0 to n-2 d
  for j<-0 to n-2-i do
       if a[j]> a[j+1] then
          exchange a[j] with a[j+1]


Program using Java
++++++++++++++++++
import java.io.*;
import java.util.*;

//Client Code
public class Test
{
   public static void main(String[] args)throws IOException
      {

         Scanner scan = new Scanner(System.in);
```

```
        System.out.print("Enter the size of an Array :: ");
        int n = scan.nextInt();

         //Create an Array of size n
        int[] arr = new int[n];

        //Fill the array elements from the user
        for (int i = 0;i<=n-1;i++ )
        {
            System.out.print("Enter the array element :: ");
            arr[i] = scan.nextInt();
        }

        System.out.println("Array before Sorting :: "+Arrays.toString(arr));

        bubbleSort(arr,n);

        System.out.println("Array before Sorting :: "+Arrays.toString(arr));

    }
    public static void bubbleSort(int[] arr,int n){
          //logic of bubbleSort
          for (int i =0; i<=n-2; i++ )
          {
                for (int j=0;j<=n-2-i ; j++)
                {
                      if (arr[j] > arr[j+1])
                      {
                            int temp = arr[j];
                            arr[j]   = arr[j+1];
                            arr[j+1] = temp;

                      }
                }
          }
    }
}
```