Name:	Debjit Ghosal
UID:	20233000065
Experiment No.	8B

AIM:	Interfaces
Program 1	
PROBLEM STATEMENT :	Write a Java program to create an interface Sortable with a method sort() that sorts an array of integers in ascending order. Create two classes BubbleSort and SelectionSort that implement the Sortable interface and provide their own implementations of the sort() method.
PROGRAM:	

```java
package Abstraction.Interfaces;

import java.util.Arrays;

interface Sortable {
    void sort(int[] arr);
}

class BubbleSort implements Sortable {

    @Override
    public void sort(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            for (int j = 0; j < arr.length-1; j++) {
                if(arr[j] > arr[j+1]) {
                    // swap.
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
            }
        }
    }
}

class SelectionSort implements Sortable {
```

```
    @Override
    public void sort(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            int minIdx = i;
            for (int j = i+1; j < arr.length; j++) {
                //find min.
                if(arr[minIdx] > arr[j]) {
                    minIdx = j;
                }
            }
            // swap the min index with proper i,
            int temp = arr[i];
            arr[i] = arr[minIdx];
            arr[minIdx] = temp;

            for(int k = 0; k < arr.length; k++){
                System.out.print(arr[k] + " ");
            }
            System.out.println(" ");
        }
    }
}

public class Sorting {
    public static void main(String[] args) {
        int[] arr = {5,1,2,3,4};
        SelectionSort selectionSort = new SelectionSort();
        System.out.println("Initial array: ");
        System.out.println(Arrays.toString(arr));


        selectionSort.sort(arr);
        System.out.println("Selection Sort: ");
        System.out.println(Arrays.toString(arr));

        BubbleSort bubbleSort = new BubbleSort();
        bubbleSort.sort(arr);
        System.out.println("Bubble sort: ");
        System.out.println(Arrays.toString(arr));
    }
}
```

RESULT:

```
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>  c:; cd 'c:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP'; & 'C:\Program Files\Java\jdk-1.8\bin\java.exe' '-cp' 'C:\Users\DEBJIT GHOSAL\AppData\Roaming\Code\User\workspaceStorage\9e74b28faa244b7149de54e209cd65d5\redhat.java\jdt_ws\PSOOP_797bc398\bin' 'Abstraction.Interfaces.Sorting'
Initial array:
[5, 1, 2, 3, 4]
1 5 2 3 4
1 2 5 3 4
1 2 3 5 4
1 2 3 4 5
1 2 3 4 5
Selection Sort:
[1, 2, 3, 4, 5]
Bubble sort:
[1, 2, 3, 4, 5]
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>
```

Program 2

PROBLEM STATEMENT :

Write a Java program to create an interface Searchable with a method search() that searches for a given word in a text document. Create two classes Document and WebPage that implement the Searchable interface and provide their own implementations of the search() method, one searches a String and other a number.

PROGRAM:

```java
package Abstraction.Interfaces;

interface Searchable {
    boolean search(String string);
}

class Document implements Searchable {
    String initialString;

    public Document(String initialString) {
        this.initialString = initialString;
    }

    @Override
    public boolean search(String string) {
        return initialString.contains(string);
    }
}

class WebPage implements Searchable {
```

```
    String initialString;

    public WebPage(String initialString) {
        this.initialString = initialString;
    }

    @Override
    public boolean search(String string) {
        return initialString.contains(string);
    }
}

public class Searching {
    public static void main(String[] args) {
        System.out.println("Document class - object.");
        String s1 = "Samarth and Debjit are best friends.";
        System.out.println("String to be search");
        Document document = new Document(s1);
        System.out.println(document.search("are good"));

        System.out.println("WebPage class - object.");
        WebPage webPage = new WebPage("This is a sample webpage
for Searching programming. No is 23");
        System.out.println(webPage.search("25"));
    }
}
```


RESULT:

```
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>  c:; cd 'c:\Users\DEBJIT GHOS
L\Desktop\SPIT_CODING\PSOOP\PSOOP'; & 'C:\Program Files\Java\jdk-1.8\bin\java.exe' '-cp'
'C:\Users\DEBJIT GHOSAL\AppData\Roaming\Code\User\workspaceStorage\9e74b28faa244b7149de5
e209cd65d5\redhat.java\jdt_ws\PSOOP_797bc398\bin' 'Abstraction.Interfaces.Searching'
Document class - object.
String to be search
true
WebPage class - object.
false
PS C:\Users\DEBJIT GHOSAL\Desktop\SPIT_CODING\PSOOP\PSOOP>
```

CONCLUSION:	I have learnt about interfaces using abstract classes.