PRACTICAL NO . 07

A. Design a generic class MyArray with add(), grow() and swap().add() adds the element to the array at last index. grow() increases the internal size by 1.5 times. grow() should be called within the add() when the size is exhausted. Declare grow() as private. swap() swaps elements the array given two indices.

Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */


/**
 *
 * @author ACER
 */

import java.util.*;
public class Niharika {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
          System.out.println("Integer\n");
```

```
 MyArray<Integer> arr = new MyArray<>();
  for(int i=0;i<10;i++){
    arr.add(i*10+1);

  }
    System.out.println("\n\n");

  for(int i=0;i<10;i++){
    System.out.println("index:" +(i+1)+"  "+arr.get(i));

  }
  System.out.println("\n\nDouble\n");
  MyArray<Double> arr1 = new MyArray<>();
  for(int i=0;i<10;i++){
    arr1.add((double)i*10+1);

  }
    System.out.println("\n\n");

  for(int i=0;i<10;i++){
    System.out.println("index:" +(i+1)+"  "+arr1.get(i));

  }

  arr1.swap(3,7);
    System.out.println("After swapping\n");
for(int i=0;i<10;i++){
    System.out.println("index:" +(i+1)+"  "+arr1.get(i));

  }

System.out.println("\n\nString\n");
  MyArray<String> arr2= new MyArray<>();
  for(int i=0;i<10;i++){
```

```
            arr2.add("String arraylist");

        }
        System.out.println("\n\n");
        for(int i=0;i<10;i++){
            System.out.println("index:" +(i+1)+"  "+arr2.get(i));

        }

    }

}
```

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author ACER
 */


class MyArray<T>{
    Object[] arr;
    int indx;
    int size;

    MyArray(T obj[]){
        size=5;
```

```
    arr=obj;
    indx=0;
  }

  MyArray(){
    size=5;
    arr = new Object[size];
    indx=0;
  }

  MyArray(int s){
    size=s;
    arr = new Object[size];
    indx=0;
  }


  void add(T item){
    if(indx<size){
      arr[indx++]=item;
      System.out.println("Item added successfully");
    }else{
      this.grow();
      this.add(item);
    }
  }

private    void grow(){
      size=size+(int)(size*0.5);
      arr=Arrays.copyOf(arr,size);
  }


  public T get(int i){
```

```
        if(i<0 || i>size-1){
            System.out.println("Exeption:Invalid index");

        }

        return (T)arr[i];
    }

    public void swap(int i,int j){
        if(i>indx||j>indx){
            System.out.println("Exception:Invalid indices");
    }else{

        Object temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;
    }
    }

}


/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author ACER
 */
```

```
class MyArray<T>{
    Object[] arr;
    int indx;
    int size;

    MyArray(T obj[]){
       size=5;
       arr=obj;
       indx=0;
    }

     MyArray(){
       size=5;
       arr = new Object[size];
       indx=0;
    }

      MyArray(int s){
       size=s;
       arr = new Object[size];
       indx=0;
    }


      void add(T item){
       if(indx<size){
          arr[indx++]=item;
          System.out.println("Item added successfully");
       }else{
          this.grow();
          this.add(item);
       }
      }
```

```
  private     void grow(){
        size=size+(int)(size*0.5);
        arr=Arrays.copyOf(arr,size);
    }


    public T get(int i){

        if(i<0 || i>size-1){
            System.out.println("Exeption:Invalid index");

        }

        return (T)arr[i];
    }

    public void swap(int i,int j){
        if(i>indx||j>indx){
            System.out.println("Exception:Invalid indices");
    }else{

        Object temp=arr[i];
        arr[i]=arr[j];
        arr[j]=temp;
    }
    }

}
```

Output:

```
Integer

Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully



index:1   1
index:2   11
index:3   21
index:4   31
index:5   41
index:6   51
index:7   61
index:8   71
index:9   81
index:10   91


Double

Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
```

```
index:1  1.0
index:2  11.0
index:3  21.0
index:4  31.0
index:5  41.0
index:6  51.0
index:7  61.0
index:8  71.0
index:9  81.0
index:10  91.0
After swapping

index:1  1.0
index:2  11.0
index:3  21.0
index:4  71.0
index:5  41.0
index:6  51.0
index:7  61.0
index:8  31.0
index:9  81.0
index:10  91.0


String

Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully
Item added successfully


index:1  String arraylist
index:2  String arraylist
index:3  String arraylist
index:4  String arraylist
index:5  String arraylist
index:6  String arraylist
index:7  String arraylist
index:8  String arraylist
index:9  String arraylist
index:10  String arraylist

[Program finished]
```

B. Design an application for maintaining a Student Phone Directory. Student has attributes like roll, name, semester,city, contact, etc. The phone directory will maintain a sorted collection of Student objects based on the semester (and roll number, if semester is same),and have functionality to add Student, remove Student, view city wise Students and view all students. if a student has missing data, then addition of such student should throw a user defined exception. Write code to demonstrate the working of all classes.

Code:

```
import java.util.*;

class Student implements Comparable<Student> {
   Integer roll_no, sem;
   String name, city;
   String ph_no;

   public int compareTo(Student ob) {
      int cmp = sem.compareTo(ob.sem);
      if (cmp == 0) {
         cmp = roll_no.compareTo(ob.roll_no);
      }
      return cmp;
   }

   Student(Integer r, Integer s, String n, String c, String p) {
      roll_no = r;
      sem = s;
      name = n;
```

```
        city = c;
        ph_no = p;
    }
}

class MissingDetailsException extends Exception {
    public MissingDetailsException(String message) {
        super(message);
    }
}

public class Main {
    public static void main(String[] args) {
        // Create some Student objects
        TreeSet<Student> students = new TreeSet<>();
        Scanner sc = new Scanner(System.in);
        int i = 0;
        while (i < 5) {
            try {
                System.out.println("Enter Name:");
                String n = sc.next();
                if (n.isEmpty()) {
                    throw new MissingDetailsException("Name is missing");
                }

                System.out.println("Enter City:");
                String c = sc.next();
                if (c.isEmpty()) {
                    throw new MissingDetailsException("City is missing");
                }

                System.out.println("Enter Phone Number:");
                String p = sc.next();
                if (p.isEmpty()) {
```

```
                throw new MissingDetailsException("Phone Number is missing");
            }

            System.out.println("Enter Roll No:");
            Integer r = sc.nextInt();

            System.out.println("Enter Semester:");
            Integer sem = sc.nextInt();

            students.add(new Student(r, sem, n, c, p));
            System.out.println("Element added successfully\n\n");
            i++;
        } catch (MissingDetailsException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

    System.out.println("Displaying students");
    display(students);

    System.out.println("\nDisplaying students city-wise");
    displayCityWise(students);
  }

  public static void display(TreeSet<Student> students) {
    System.out.println("Rollno\tName\t\tCity\t\tSemester\tPhone No");
    for (Student student : students) {
      System.out.printf("%-6d\t%-15s\t%-15s\t%-8d\t%s%n",
          student.roll_no, student.name, student.city, student.sem,
student.ph_no);
    }
  }

  public static void displayCityWise(TreeSet<Student> students) {
```

```
        TreeSet<String> cities = new TreeSet<>();
        for (Student student : students) {
            cities.add(student.city);
        }

        for (String city : cities) {
            System.out.println("City: " + city);

            for (Student student : students) {
                if (student.city.equals(city)) {
                    System.out.printf("%-6d\t%-15s\t%-8d\t%s%n",
                            student.roll_no, student.name, student.sem, student.ph_no);
                }
            }

            System.out.println();
        }
    }
}
```

Output:

```
Enter Name:
Niharika
Enter City:
Nagpur
Enter Phone Number:
988778
Enter Roll No:
19
Enter Semester:
4
Element added successfully


Enter Name:
Nikki
Enter City:
Nagpur
Enter Phone Number:
788999
Enter Roll No:
8
```

```
Enter Name:
Nikki
Enter City:
Nagpur
Enter Phone Number:
788999
Enter Roll No:
8
Enter Semester:
6
Element added successfully


Enter Name:
Niku
Enter City:
Bhandara
Enter Phone Number:
788857
Enter Roll No:
```

```
Enter Roll No:
65
Enter Semester:
4
Element added successfully


Enter Name:
Bhola
Enter City:
Ramtek
Enter Phone Number:
687567
Enter Roll No:
56
Enter Semester:
6
Element added successfully


Enter Name:
```

```
Displaying students
Rollno  Name          City          Semester      Phone No
19      Niharika      Nagpur        4             988778
65      Niku          Bhandara      4             788857
8       Nikki         Nagpur        6             788999
56      Bhola         Ramtek        6             687567
65      Nick          Ramtek        6             978756

Displaying students city-wise
City: Bhandara
65      Niku          4             788857

City: Nagpur
19      Niharika      4             988778
8       Nikki         6             788999

City: Ramtek
56      Bhola         6             687567
65      Nick          6             978756
```

Result:Hence in this practical I have successfully implemented programs using generics.