KHAN MOHD OWAIS RAZA
20BCD7138

Q.1]

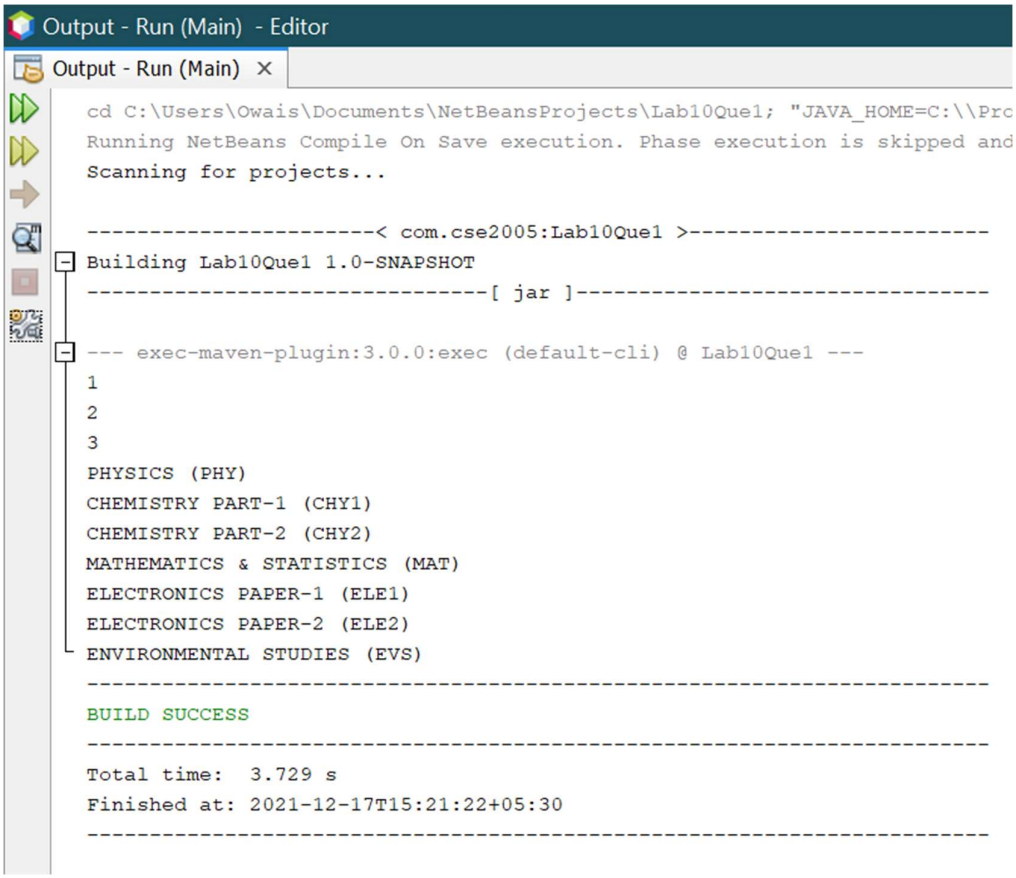
Main.java - Editor

Main.java

Source History

```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-10 */
package com.cse2005.lab10que1;
import java.lang.reflect.Method;
class Printer {
    public <inputType> void printArray(inputType[] array) {
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i]);
        }
    }
}
public class Main {
    public static void main(String[] args) {
        Printer myPrinter = new Printer();
        Integer[] intArray = { 1, 2, 3 };
        String[] stringArray = {"PHYSICS (PHY)",
                                "CHEMISTRY PART-1 (CHY1)",
                                "CHEMISTRY PART-2 (CHY2)",
                                "MATHEMATICS & STATISTICS (MAT)",
                                "ELECTRONICS PAPER-1 (ELE1)",
                                "ELECTRONICS PAPER-2 (ELE2)",
                                "ENVIRONMENTAL STUDIES (EVS)"};
        myPrinter.printArray( intArray );
        myPrinter.printArray( stringArray );
        int printArrayMethodCount = 0;
        for (Method method : Printer.class.getDeclaredMethods()) {
            String methodName = method.getName();
            if (methodName.equals("printArray")) {
                printArrayMethodCount++;
            }
        }
        if (printArrayMethodCount > 1) {
            System.out.println("Method overloading is not allowed!");
            assert printArrayMethodCount == 1;
        }
    }
}
```

Output console –



Q.2]

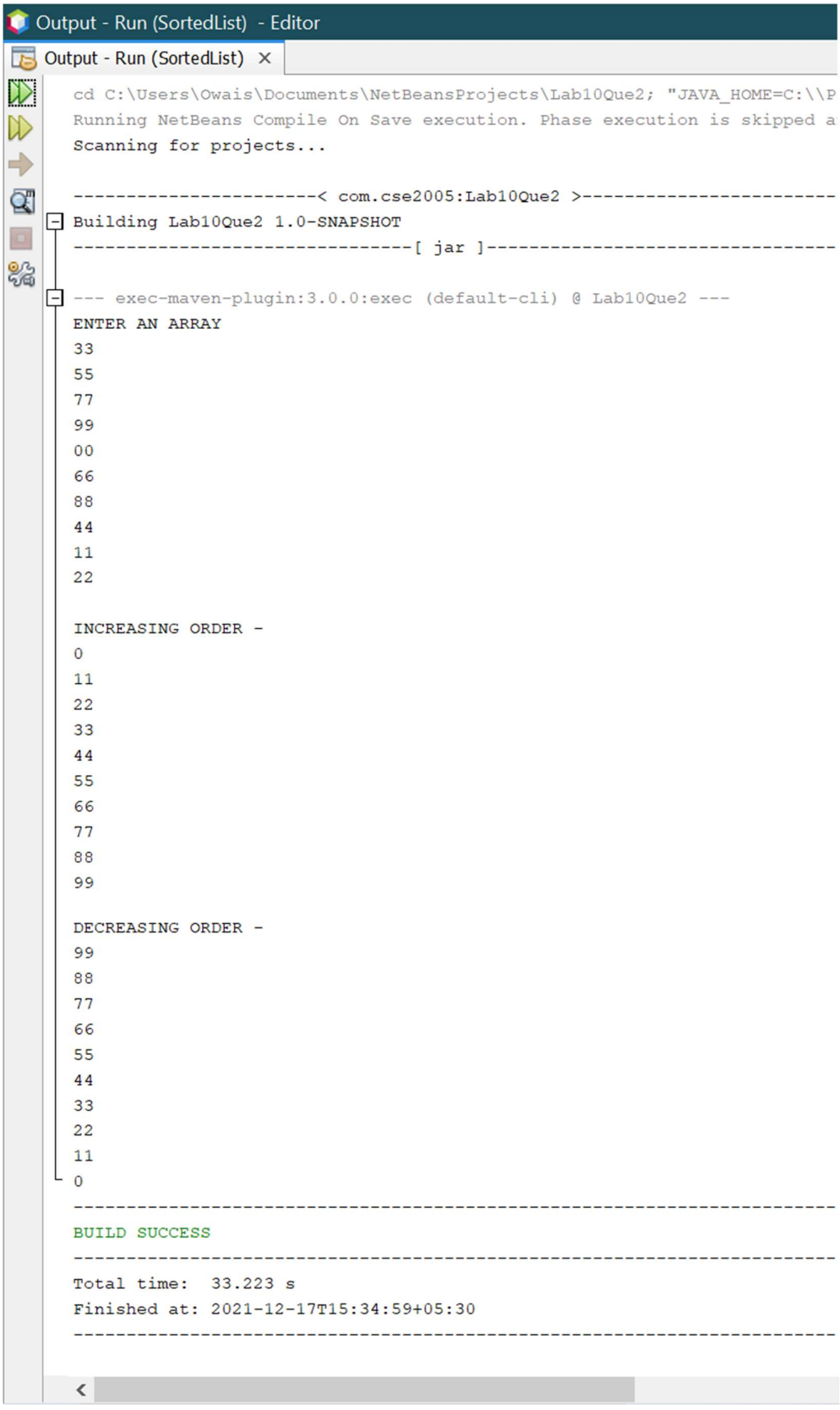
SortedList.java - Editor

SortedList.java

Source | History

```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-10 */
package com.cse2005.lab10que2;
import java.util.Scanner;
public class SortedList {
    public static void main(String[] args) {
        Scanner ed = new Scanner(System.in);
        int[] a = new int[10];
        int i, j, temp;
        System.out.println("ENTER AN ARRAY");
        for (i = 0; i < 10; i++) {
            a[i] = ed.nextInt();
        }
        for (i = 0; i < 10; i++) {
            for (j = i + 1; j < 10; j++) {
                if (a[i] > a[j]) {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.println("\nINCREASING ORDER -");
        for (j = 0; j < 10; j++) {
            System.out.println(a[j]);
        }
        for (i = 0; i < 10; i++) {
            for (j = i + 1; j < 10; j++) {
                if (a[i] < a[j]) {
                    temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
            }
        }
        System.out.println("\nDECREASING ORDER -");
        for (j = 0; j < 10; j++) {
            System.out.println(a[j]);
        }
    }
}
```

Output console –



Q.3]

Main.java - Editor

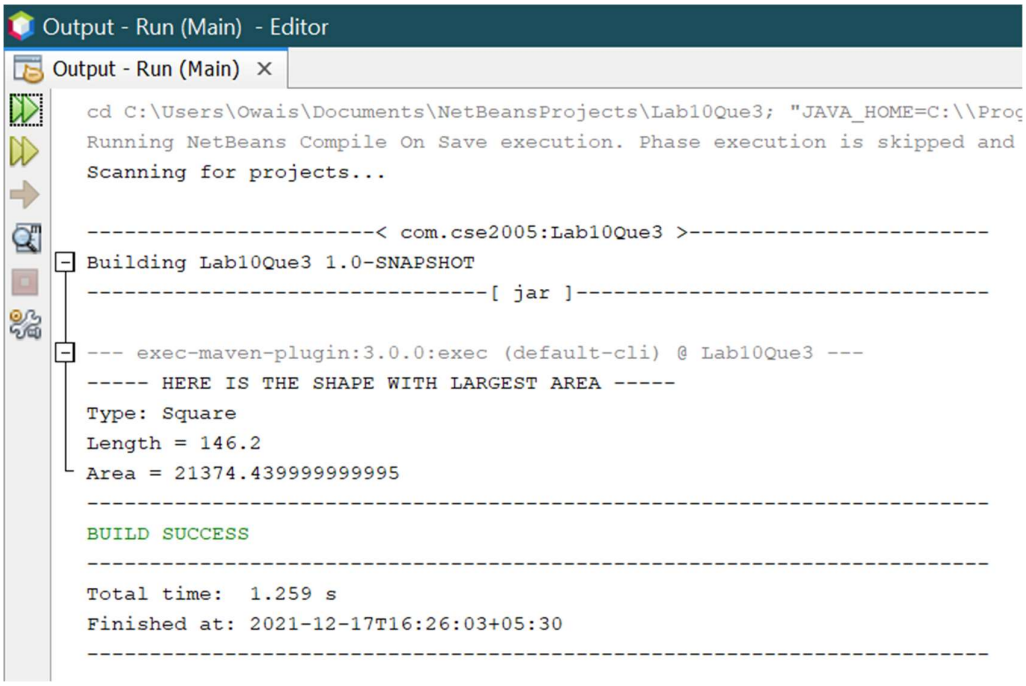
Main.java

Source | History

```java
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-10 */
package com.cse2005.lab10que3;
import java.util. * ;
abstract class Node{
    private double area;
    abstract void calcArea();
    abstract void DisplayDetails();
    public double getArea(){
        return area;
    }
public void setArea(double ar){
    area = ar;
}
public static void compareArea(List < ?>list){
    Node largest = (Node)list.get(0);
    for(int i = 0; i < list.size(); i++){
        if(((Node)list.get(i)).getArea() > largest.getArea()){
            largest = (Node)list.get(i);
        }
    }
    System.out.println("----- HERE IS THE SHAPE WITH LARGEST AREA -----" );
    largest.DisplayDetails();
}
}
class Circle extends Node{
    double radius;
    Circle(double radius)
    {
        this.radius = radius;
        calcArea();
    }
    void calcArea()
    {
        setArea(3.14 * radius * radius);
    }
    void DisplayDetails(){
        System.out.println("Type: Circle");
        System.out.println("Radius = " + radius);
        System.out.println("Area = " + getArea());
    }
}
```

```
43      class Square extends Node{
44          double length;
45          Square(double length){
                this.length = length;calcArea();
47          }
            void calcArea(){
49              setArea(length * length);
50          }
            void DisplayDetails(){
52              System.out.println("Type: Square");
53              System.out.println("Length = " + length);
54              System.out.println("Area = " + getArea());
55          }
56      }
57      class Rectangle extends Node{
58          double l;
59          double b;
60          Rectangle(double l, double b){
61              this.l = l;
62              this.b = b;
                calcArea();
64          }
            void calcArea(){
66              setArea( l * b);
67          }
            void DisplayDetails(){
69              System.out.println("Type: Rectangle");
70              System.out.println("Length = " + l);
71              System.out.println("Breadth = " + b);
72              System.out.println("Area = " + getArea());
73          }
74      }
75      public class Main{
76          public static void main(String[] args){
77              List<Node> ls=new ArrayList<>();
78              ls.add(new Rectangle(30, 40));
79              ls.add(new Circle(4.0));
80              ls.add(new Square(146.2));
                ls.get(0).compareArea(ls);
82          }
83      }
```

Output console –



Q.4]

```
/* KHAN MOHD OWAIS RAZA 20BCD7138 */
/* CSE2005 LAB-10 */
package com.cse2005.lab10q4;
import java.util.Arrays;
import java.util.HashSet;
import java.util.Set;
class CSE2005_Lab10_Qu4 {
    void printUnion(int arr1[], int arr2[], int m, int n){
        if (m > n) {
            int tempp[] = arr1;
            arr1 = arr2;
            arr2 = tempp;
            int temp = m;
            m = n;
            n = temp;
        }
        Arrays.sort(arr1);
        for (int i = 0; i < m; i++)
            System.out.print(arr1[i] + " ");
        for (int i = 0; i < n; i++) {
            if (binarySearch(arr1, 0, m - 1, arr2[i]) == -1)
                System.out.print(arr2[i] + " ");
        }
    }
    void printIntersection(int arr1[], int arr2[], int m, int n){
        if (m > n) {
            int tempp[] = arr1;
            arr1 = arr2;
            arr2 = tempp;
            int temp = m;
            m = n;
            n = temp;
        }
```

```
    Arrays.sort(arr1);
    for (int i = 0; i < n; i++) {
        if (binarySearch(arr1, 0, m - 1, arr2[i]) != -1)
            System.out.println(arr2[i] + " ");
    }
}
int binarySearch(int arr[], int l, int r, int x){
    if (r >= l) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] > x)
            return binarySearch(arr, l, mid - 1, x);
                return binarySearch(arr, mid + 1, r, x);
         }
    return -1;
}



     boolean ji = true;
     boolean ji1 = true;

    boolean check(int arr1[],int ch)
    {
         for(int i =0;i<arr1.length;i++)
         {
                if(ch==arr1[i])
                {

                       ji= true;
                }
                else
                {
```

```
                        ji= false;
                }
        }
        return ji;
    }
    boolean check1(int arr2[],int ch)
    {
        for(int i =0;i<arr2.length;i++)
        {
            if(ch==arr2[i])
            {

                ji1= true;
            }
            else
            {

                ji1= false;
            }
        }
        return ji1;
    }

}
class Boolean_Operation
{
    public static void main(String[] args){
        CSE2005_Lab10_Qu4 u_i;
        u_i = new CSE2005_Lab10_Qu4();

        int arr1[] = { 7, 1, 5, 2, 3, 6 };
        int arr2[] = { 3, 8, 6, 2, 1, 7 };
        int m = arr1.length;
        int n = arr2.length;
```

```
            System.out.println("Union of two arrays is ");
            u_i.printUnion(arr1, arr2, m, n);
            System.out.println("");
            System.out.println("Intersection of two arrays is ");
            int nw = 7;
            u_i.printIntersection(arr1, arr2, m, n);
            boolean jh;
           jh= u_i.check(arr1, 7);
           boolean jh1;
           jh1= u_i.check(arr1, 7);

      if(jh==true)
      {
           System.out.println("True found in Arr1");
      }
      if(jh1==true)
      {
           System.out.println("True found  in Arr2");
      }
      }
}
```

Output –

```
Union of two arrays is
1 2 3 5 6 7 8
Intersection of two arrays is
3
6
2
1
7
True found in Arr1
True found  in Arr2
```