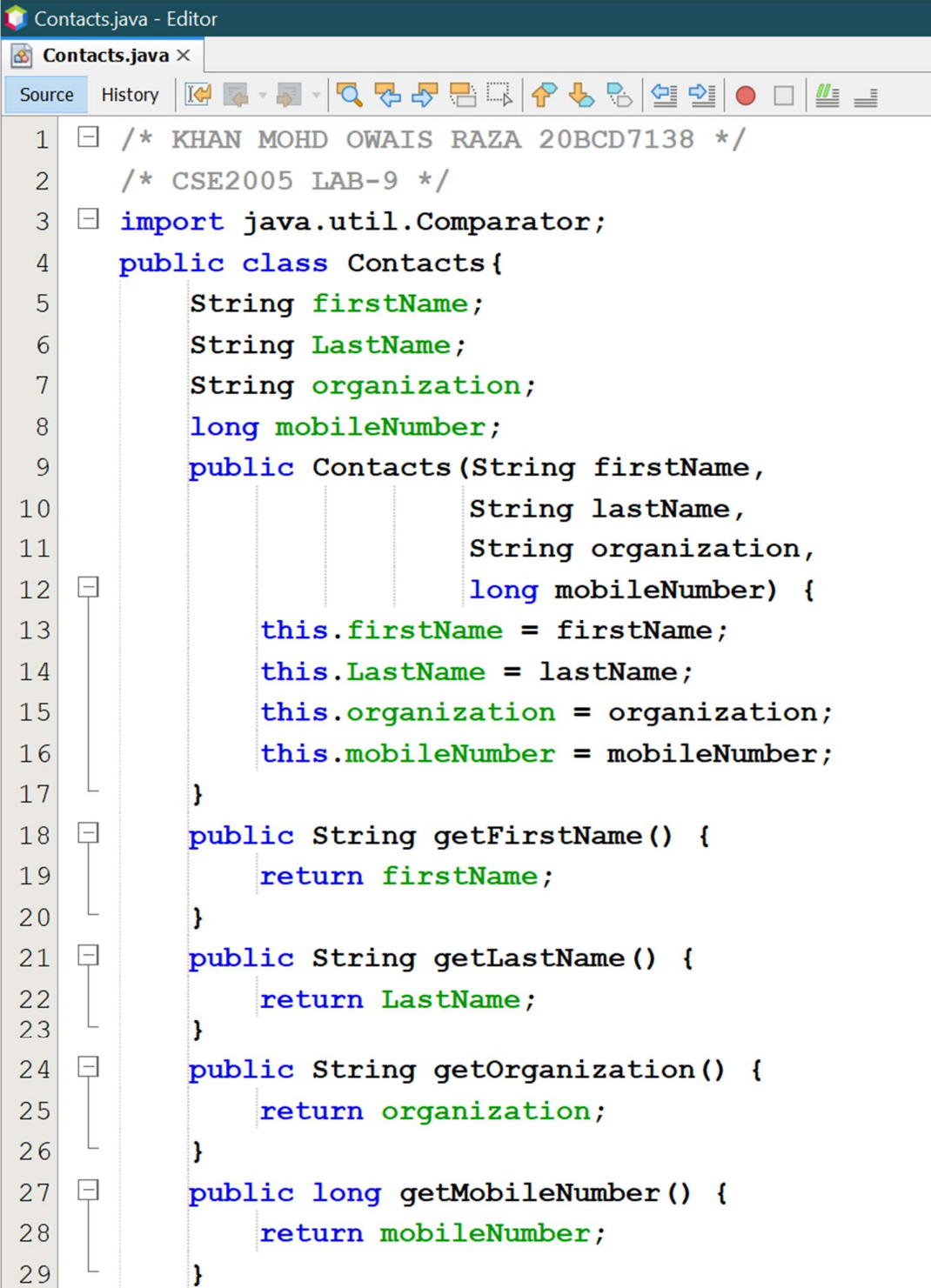


KHAN MOHD. OWAIS RAZA
20BCD7138

Q.1] Develop a class “Contacts” with attributes first name, last name, organization, mobile number. Develop another class “ContactList” to hold all contacts of a user in the form of Contacts class objects and as an Array List. Now the user wants to display all the contacts in ascending order of first name or descending order of last name. Develop a suitable logic using Comparator interface to display contacts in requested order.

Contacts.java –



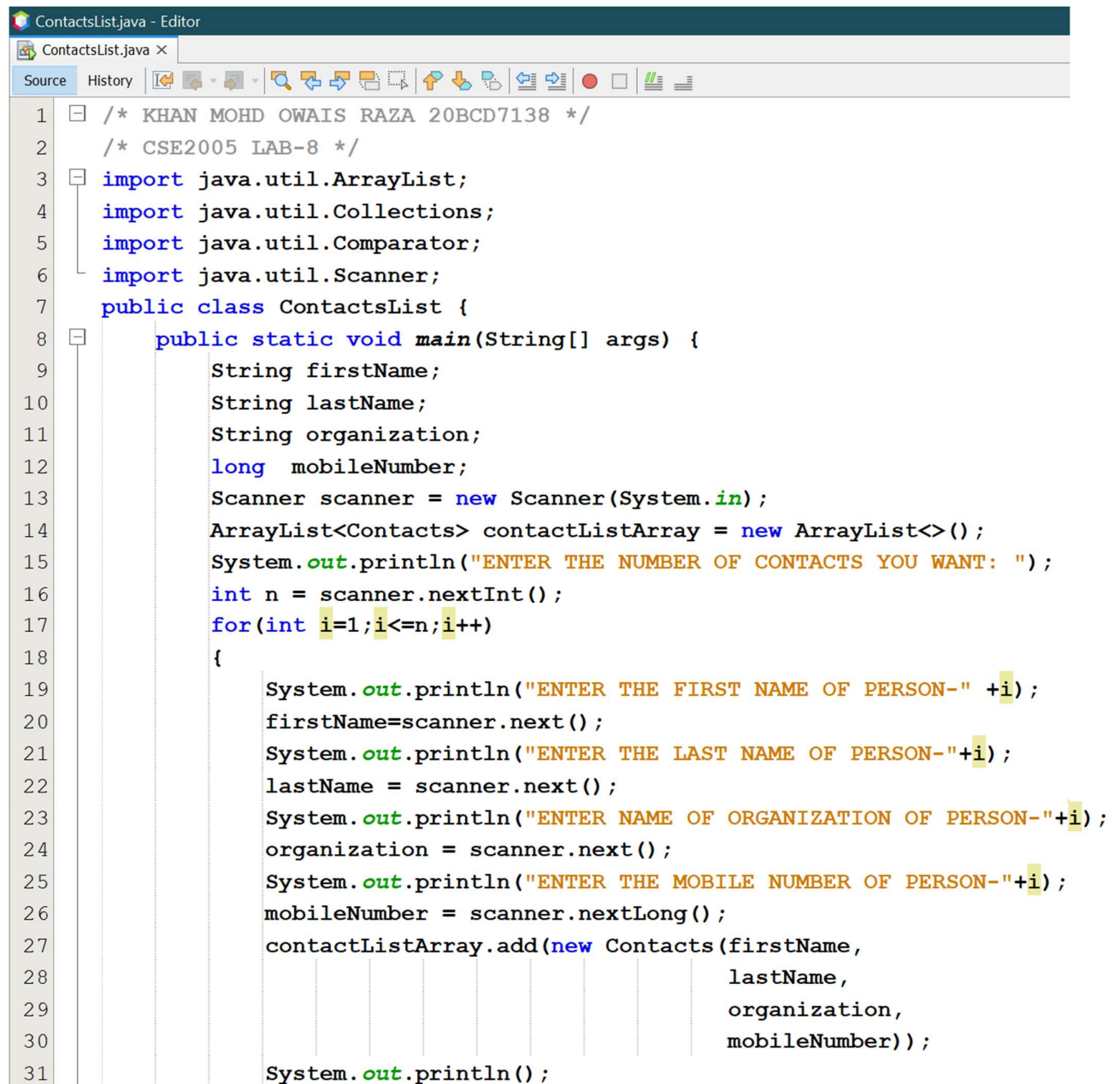
```
Contacts.java - Editor
Contacts.java x
Source History
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-9 */
3  import java.util.Comparator;
4  public class Contacts{
5      String firstName;
6      String lastName;
7      String organization;
8      long mobileNumber;
9      public Contacts(String firstName,
10                      String lastName,
11                      String organization,
12                      long mobileNumber) {
13          this.firstName = firstName;
14          this.lastName = lastName;
15          this.organization = organization;
16          this.mobileNumber = mobileNumber;
17      }
18      public String getFirstName() {
19          return firstName;
20      }
21      public String getLastName() {
22          return lastName;
23      }
24      public String getOrganization() {
25          return organization;
26      }
27      public long getMobileNumber() {
28          return mobileNumber;
29      }
```

```

30         @Override
31         public String toString() {
32             return
33                 "\nFirst Name: " + firstName +
34                 "\nLast Name: " + LastName +
35                 "\nOrganization: " + organization +
36                 "\nMobile Number: " + mobileNumber;
37         }
38     }
39     class FirstNameSorter implements Comparator<Contacts>{
40         @Override
41         public int compare(Contacts o1, Contacts o2) {
42             return o1.getFirstName().compareTo(o2.getFirstName());
43         }
44     }

```

ContactsList.java –



```

1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-8 */
3  import java.util.ArrayList;
4  import java.util.Collections;
5  import java.util.Comparator;
6  import java.util.Scanner;
7  public class ContactsList {
8      public static void main(String[] args) {
9          String firstName;
10         String lastName;
11         String organization;
12         long mobileNumber;
13         Scanner scanner = new Scanner(System.in);
14         ArrayList<Contacts> contactListArray = new ArrayList<>();
15         System.out.println("ENTER THE NUMBER OF CONTACTS YOU WANT: ");
16         int n = scanner.nextInt();
17         for(int i=1;i<=n;i++)
18         {
19             System.out.println("ENTER THE FIRST NAME OF PERSON-" +i);
20             firstName=scanner.next();
21             System.out.println("ENTER THE LAST NAME OF PERSON-"+i);
22             lastName = scanner.next();
23             System.out.println("ENTER NAME OF ORGANIZATION OF PERSON-"+i);
24             organization = scanner.next();
25             System.out.println("ENTER THE MOBILE NUMBER OF PERSON-"+i);
26             mobileNumber = scanner.nextLong();
27             contactListArray.add(new Contacts(firstName,
28                                             lastName,
29                                             organization,
30                                             mobileNumber));
31             System.out.println();

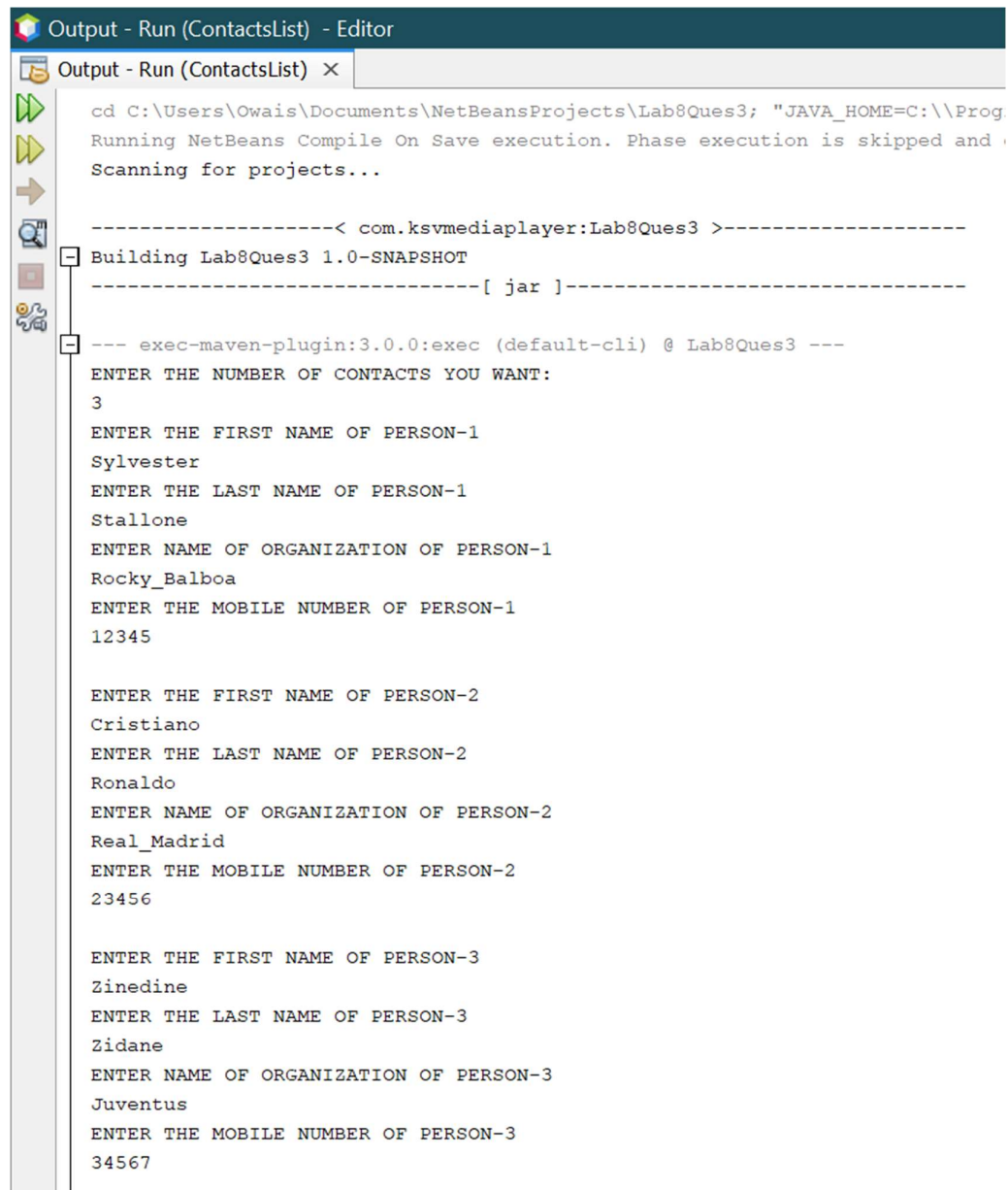
```

```

32     }
33     System.out.println("PRINTING OF UNSORTED PASSED ARRAY LIST");
34     System.out.println(contactListArray);
35     System.out.println("PRINTING LIST IN ASCENDING ORDER OF FIRST NAME");
36     Collections.sort(contactListArray, new FirstNameSorter());
37     System.out.println(contactListArray);
38     System.out.println("PRINTING LIST IN DESCENDING ORDER OF LAST NAME: ");
39     Comparator<Contacts> comparator
40         = Comparator.comparing(e -> e.getLastName());
41     contactListArray.sort(comparator.reversed());
42     System.out.println(contactListArray);
43 }
44 }

```

Output console –



```

Output - Run (ContactsList) - Editor
Output - Run (ContactsList) x
cd C:\Users\Owais\Documents\NetBeansProjects\Lab8Ques3; "JAVA_HOME=C:\\Prog
Running NetBeans Compile On Save execution. Phase execution is skipped and
Scanning for projects...

-----< com.ksvmediaplayer:Lab8Ques3 >-----
Building Lab8Ques3 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab8Ques3 ---
ENTER THE NUMBER OF CONTACTS YOU WANT:
3
ENTER THE FIRST NAME OF PERSON-1
Sylvester
ENTER THE LAST NAME OF PERSON-1
Stallone
ENTER NAME OF ORGANIZATION OF PERSON-1
Rocky_Balboa
ENTER THE MOBILE NUMBER OF PERSON-1
12345

ENTER THE FIRST NAME OF PERSON-2
Cristiano
ENTER THE LAST NAME OF PERSON-2
Ronaldo
ENTER NAME OF ORGANIZATION OF PERSON-2
Real_Madrid
ENTER THE MOBILE NUMBER OF PERSON-2
23456

ENTER THE FIRST NAME OF PERSON-3
Zinedine
ENTER THE LAST NAME OF PERSON-3
Zidane
ENTER NAME OF ORGANIZATION OF PERSON-3
Juventus
ENTER THE MOBILE NUMBER OF PERSON-3
34567

```

```
PRINTINTING OF UNSORTED PASSED ARRAY LIST
[
First Name: Sylvester
Last Name: Stallone
Organization: Rocky_Balboa
Mobile Number: 12345,
First Name: Cristiano
Last Name: Ronaldo
Organization: Real_Madrid
Mobile Number: 23456,
First Name: Zinedine
Last Name: Zidane
Organization: Juventus
Mobile Number: 34567]
PRINTING LIST IN ASCENDING ORDER OF FIRST NAME
```

```
[
First Name: Cristiano
Last Name: Ronaldo
Organization: Real_Madrid
Mobile Number: 23456,
First Name: Sylvester
Last Name: Stallone
Organization: Rocky_Balboa
Mobile Number: 12345,
First Name: Zinedine
Last Name: Zidane
Organization: Juventus
Mobile Number: 34567]
PRINTING LIST IN DESCENDING ORDER OF LAST NAME:
```

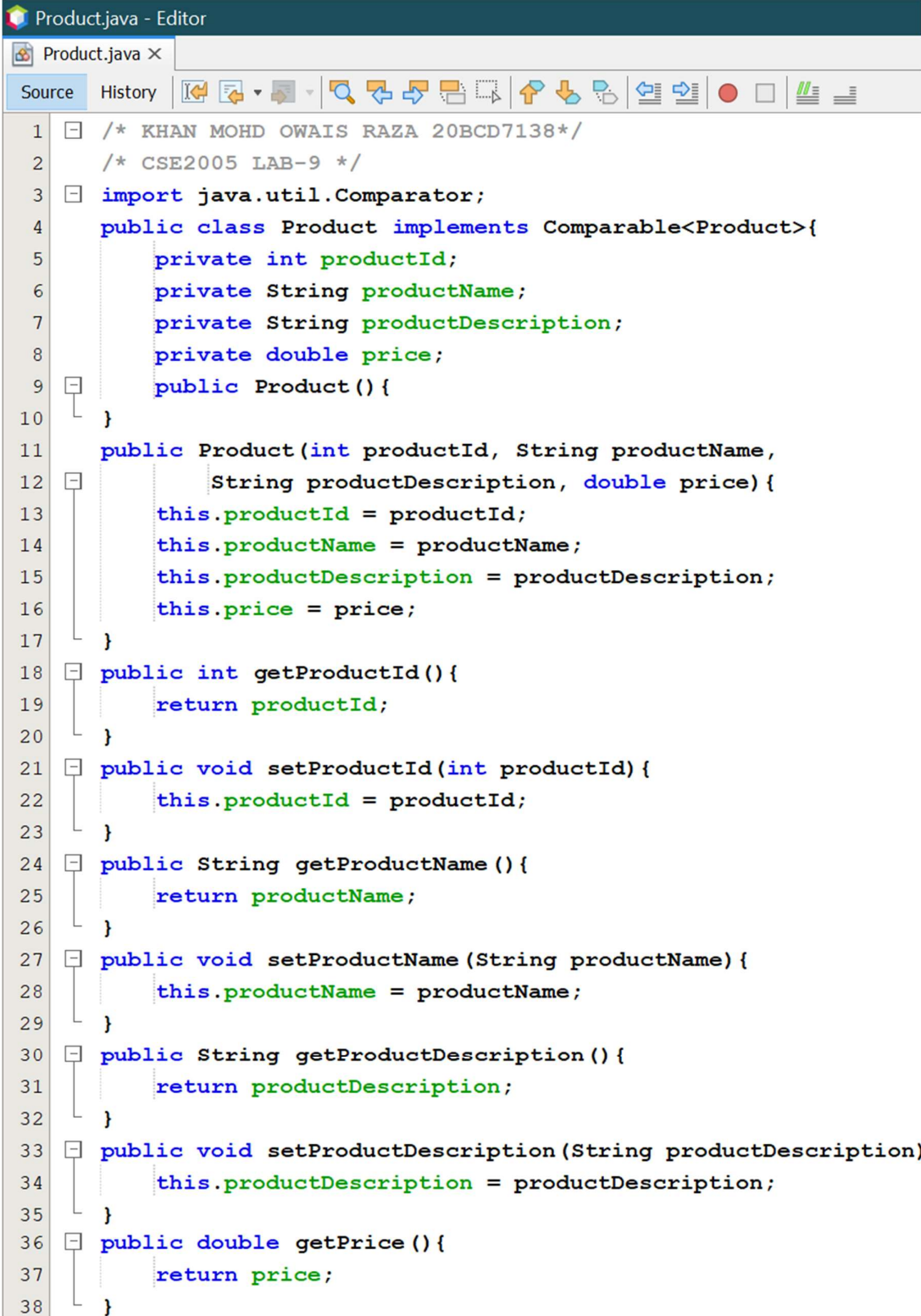
```
[
First Name: Zinedine
Last Name: Zidane
Organization: Juventus
Mobile Number: 34567,
First Name: Sylvester
Last Name: Stallone
Organization: Rocky_Balboa
Mobile Number: 12345,
First Name: Cristiano
Last Name: Ronaldo
Organization: Real_Madrid
Mobile Number: 23456]
```

```
-----
BUILD SUCCESS
-----
```

```
Total time: 02:50 min
Finished at: 2021-12-15T15:00:51+05:30
-----
```


Q.2] Develop a class “Product” with attributes product id, product name, product price. Develop another class “ProductList” to hold all products of a shopping mall as Product objects and as a Linked List. Now the store manager wants to display the products either in ascending order of product names or descending order of product price. Develop a suitable logic using Comparator interface to display products in requested order.

Product.java –



```
1  /* KHAN MOHD OWAIS RAZA 20BCD7138*/
2  /* CSE2005 LAB-9 */
3  import java.util.Comparator;
4  public class Product implements Comparable<Product>{
5      private int productId;
6      private String productName;
7      private String productDescription;
8      private double price;
9      public Product () {
10     }
11     public Product(int productId, String productName,
12     String productDescription, double price){
13         this.productId = productId;
14         this.productName = productName;
15         this.productDescription = productDescription;
16         this.price = price;
17     }
18     public int getProductId(){
19         return productId;
20     }
21     public void setProductId(int productId) {
22         this.productId = productId;
23     }
24     public String getProductName () {
25         return productName;
26     }
27     public void setProductName (String productName) {
28         this.productName = productName;
29     }
30     public String getProductDescription () {
31         return productDescription;
32     }
33     public void setProductDescription (String productDescription) {
34         this.productDescription = productDescription;
35     }
36     public double getPrice () {
37         return price;
38     }
```

```

39  public void setPrice(double price){
40      this.price = price;
41  }
42  @Override
43  public int compareTo(Product p){
44      return this.productId-p.productId;
45  }
46  @Override
47  public String toString(){
48      return "\nPrice: " + price +
49              "\nProduct Description: " + productDescription +
50              "\nProduct ID: " + productId+
51              "\nProduct Name: " + productName;
52  }
53  public static class ProductInnerClass implements Comparator<Product>
54  {
55      @Override
56      public int compare(Product p1, Product p2){
57          int i=Double.compare(p1.getPrice(), p2.getPrice());
58          if(i==0){
59              return p1.getProductid()-p2.getProductid();
60          }
61          return i;
62      }
63  }
64  }

```

ProductList.java –

```

/* KHAN MOHD OWAIS RAZA 20BCD7138*/
/* CSE2005 LAB-9 */

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;

public class ProductList {

    public static void main(String[] args) throws NumberFormatException, IOException {

        ArrayList<Product> productList=new ArrayList<Product>();

        productList.add(new Product(5555, "Sony Bravia", "LED TV (32 inch)",
120000.00));

        productList.add(new Product(2222, "Apple iMac", "Desktop PC", 85000.00));

        productList.add(new Product(3333, "Samsung Galaxy M21", "Mobile Phone",
4000.00));

        productList.add(new Product(6666, "Daikin 1.5 Ton AC", "Air Conditioner",
200.00));
    }
}

```

```

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
int i=1;
while(i<3){
    System.out.println("(1) SORT BY ID");
    System.out.println("(2) SORT BY NAME");
    System.out.println("(3) SORT BY PRICE");
    int ch=Integer.parseInt(br.readLine());
    switch(ch){
        case 1 :
            Collections.sort(productList);
            System.out.println("Sorted product : ");
            for(Product p : productList){
                System.out.println(p);
            }
            break;
        case 2 :
            Collections.sort(productList, new Comparator<Product>(){
                @Override
                public int compare(Product p1, Product p2){
                    int
i=p1.getProductName().compareToIgnoreCase(p2.getProductName());
                    if(i==0){
                        return
p1.getProductDescription().compareToIgnoreCase(p2.getProductDescription());
                    }
                    return i;
                }
            });
            System.out.println("Sorted product : ");
            for(Product p : productList){
                System.out.println(p);
            }
            break;
        case 3 :
            Collections.sort(productList,new Product.ProductInnerClass());

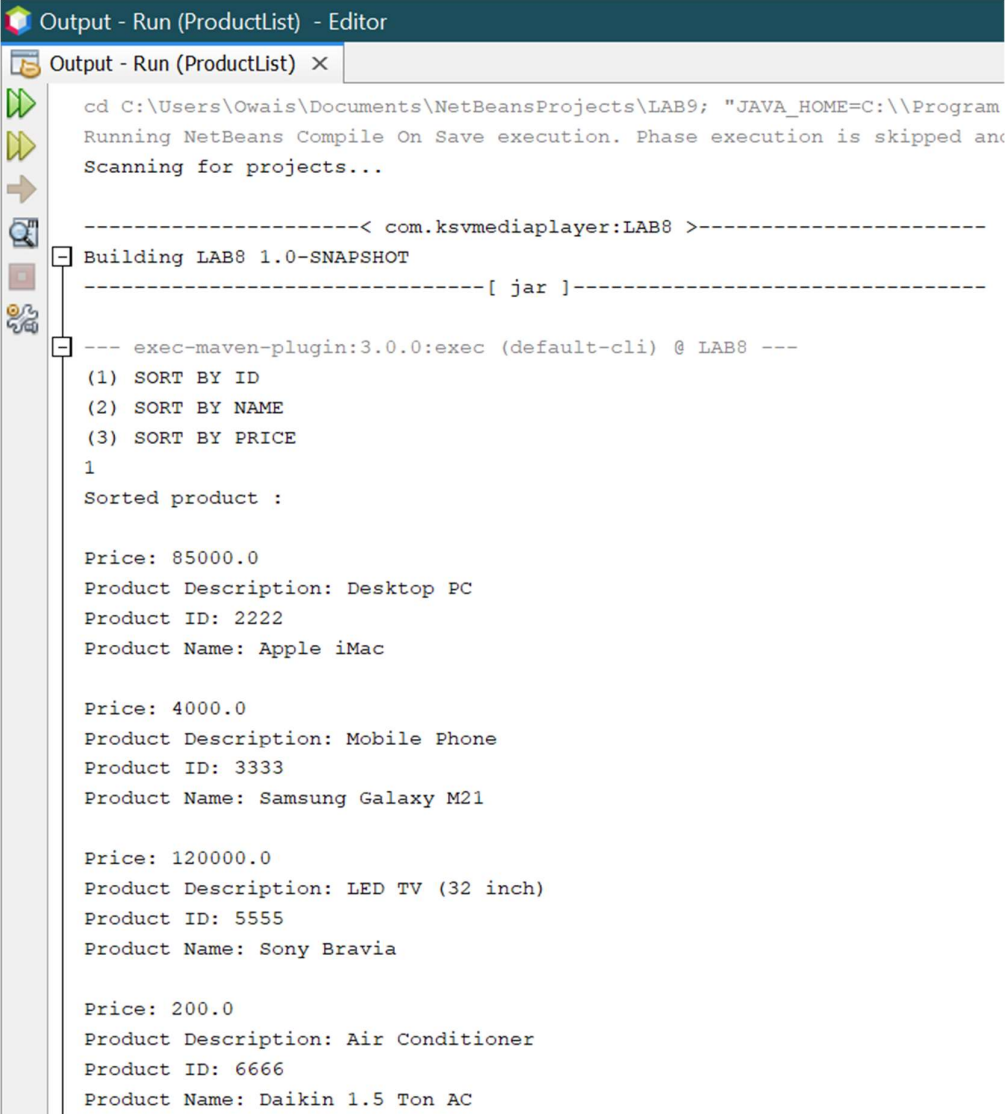
```

```

        System.out.println("Sorted product : ");
        for(Product p : productList){
            System.out.println(p);
        }
        break;
    default :
        System.out.println("Invalid Option");
        System.exit(0);
    }
}
}
}

```

Output console –



```

Output - Run (ProductList) - Editor
Output - Run (ProductList) x
cd C:\Users\Owais\Documents\NetBeansProjects\LAB9; "JAVA_HOME=C:\\Program
Running NetBeans Compile On Save execution. Phase execution is skipped and
Scanning for projects...

-----< com.ksvmediaplayer:LAB8 >-----
Building LAB8 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ LAB8 ---
(1) SORT BY ID
(2) SORT BY NAME
(3) SORT BY PRICE
1
Sorted product :

Price: 85000.0
Product Description: Desktop PC
Product ID: 2222
Product Name: Apple iMac

Price: 4000.0
Product Description: Mobile Phone
Product ID: 3333
Product Name: Samsung Galaxy M21

Price: 120000.0
Product Description: LED TV (32 inch)
Product ID: 5555
Product Name: Sony Bravia

Price: 200.0
Product Description: Air Conditioner
Product ID: 6666
Product Name: Daikin 1.5 Ton AC

```


- (1) SORT BY ID
- (2) SORT BY NAME
- (3) SORT BY PRICE

2

Sorted product :

Price: 85000.0
Product Description: Desktop PC
Product ID: 2222
Product Name: Apple iMac

Price: 200.0
Product Description: Air Conditioner
Product ID: 6666
Product Name: Daikin 1.5 Ton AC

Price: 4000.0
Product Description: Mobile Phone
Product ID: 3333
Product Name: Samsung Galaxy M21

Price: 120000.0
Product Description: LED TV (32 inch)
Product ID: 5555
Product Name: Sony Bravia

- (1) SORT BY ID
- (2) SORT BY NAME
- (3) SORT BY PRICE

3

Sorted product :

Price: 200.0
Product Description: Air Conditioner
Product ID: 6666
Product Name: Daikin 1.5 Ton AC

Price: 4000.0
Product Description: Mobile Phone
Product ID: 3333
Product Name: Samsung Galaxy M21

Price: 85000.0
Product Description: Desktop PC
Product ID: 2222
Product Name: Apple iMac

Price: 120000.0
Product Description: LED TV (32 inch)
Product ID: 5555
Product Name: Sony Bravia

- (1) SORT BY ID
- (2) SORT BY NAME
- (3) SORT BY PRICE

4

Invalid Option

BUILD SUCCESS

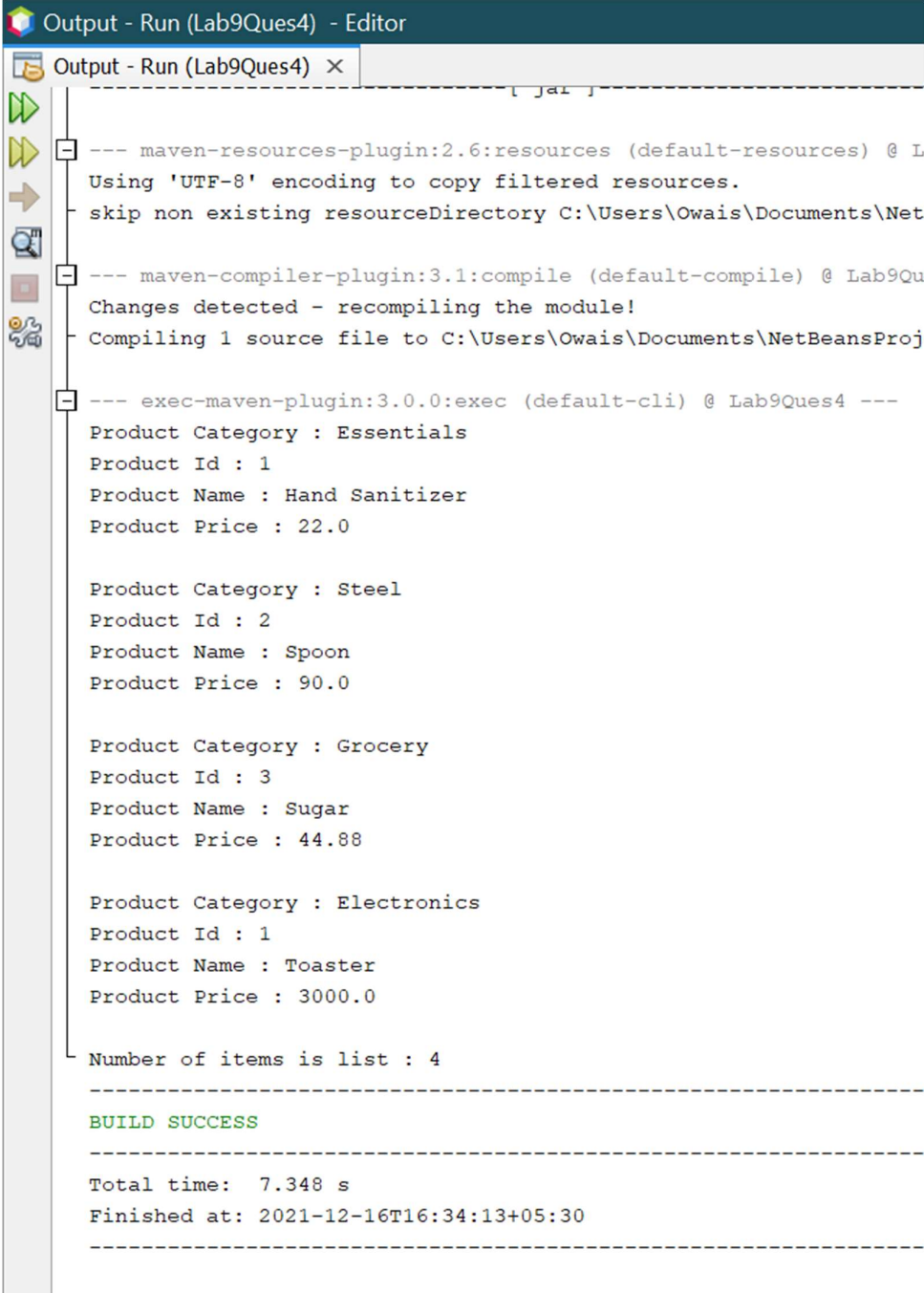
Total time: 25.577 s
Finished at: 2021-12-16T15:38:51+05:30

Q.4]

ProductList.java –

```
ProductList.java - Editor
ProductList.java x
Source History
1  /* KHAN MOHD OWAIS RAZA 20BCD7138*/
2  /* CSE2005 LAB-9 */
3  import java.util.HashMap;
4  class ProductList{
5      int size = 0;
6      HashMap<Integer, String> category = new HashMap<>();
7      HashMap<Integer, Integer> id = new HashMap<>();
8      HashMap<Integer, String> name = new HashMap<>();
9      HashMap<Integer, Double> price = new HashMap<>();
10     public void addItem(String category, int id, String name, Double price){
11         size++;
12         this.category.put(size, category);
13         this.id.put(size, id);
14         this.name.put(size, name);
15         this.price.put(size, price);
16     }
17     public void displayItems(){
18         for(int i = 1; i <= size; i++){
19             System.out.println("Product Category : " + category.get(i));
20             System.out.println("Product Id : " + id.get(i));
21             System.out.println("Product Name : " + name.get(i));
22             System.out.println("Product Price : " + price.get(i) + "\n");
23         }
24     }
25     public int itemCount(){
26         return size;
27     }
28 }
29 class Main {
30     public static void main(String[] args) {
31         ProductList p = new ProductList();
32         p.addItem("Essentials", 1, "Hand Sanitizer", 22.00);
33         p.addItem("Steel", 2, "Spoon", 90.00);
34         p.addItem("Grocery", 3, "Sugar", 44.88);
35         p.addItem("Electronics", 1, "Toaster", 3000.00);
36
37         p.displayItems();
38
39         System.out.println("Number of items is list : " + p.itemCount());
40     }
41 }
```

Output Console –



```
Output - Run (Lab9Ques4) - Editor
Output - Run (Lab9Ques4) X
--- maven-resources-plugin:2.6:resources (default-resources) @ L
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory C:\Users\Owais\Documents\Net
--- maven-compiler-plugin:3.1:compile (default-compile) @ Lab9Qu
Changes detected - recompiling the module!
Compiling 1 source file to C:\Users\Owais\Documents\NetBeansProj
--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab9Ques4 ---
Product Category : Essentials
Product Id : 1
Product Name : Hand Sanitizer
Product Price : 22.0

Product Category : Steel
Product Id : 2
Product Name : Spoon
Product Price : 90.0

Product Category : Grocery
Product Id : 3
Product Name : Sugar
Product Price : 44.88

Product Category : Electronics
Product Id : 1
Product Name : Toaster
Product Price : 3000.0

Number of items is list : 4

BUILD SUCCESS

Total time: 7.348 s
Finished at: 2021-12-16T16:34:13+05:30
```

Q.3]

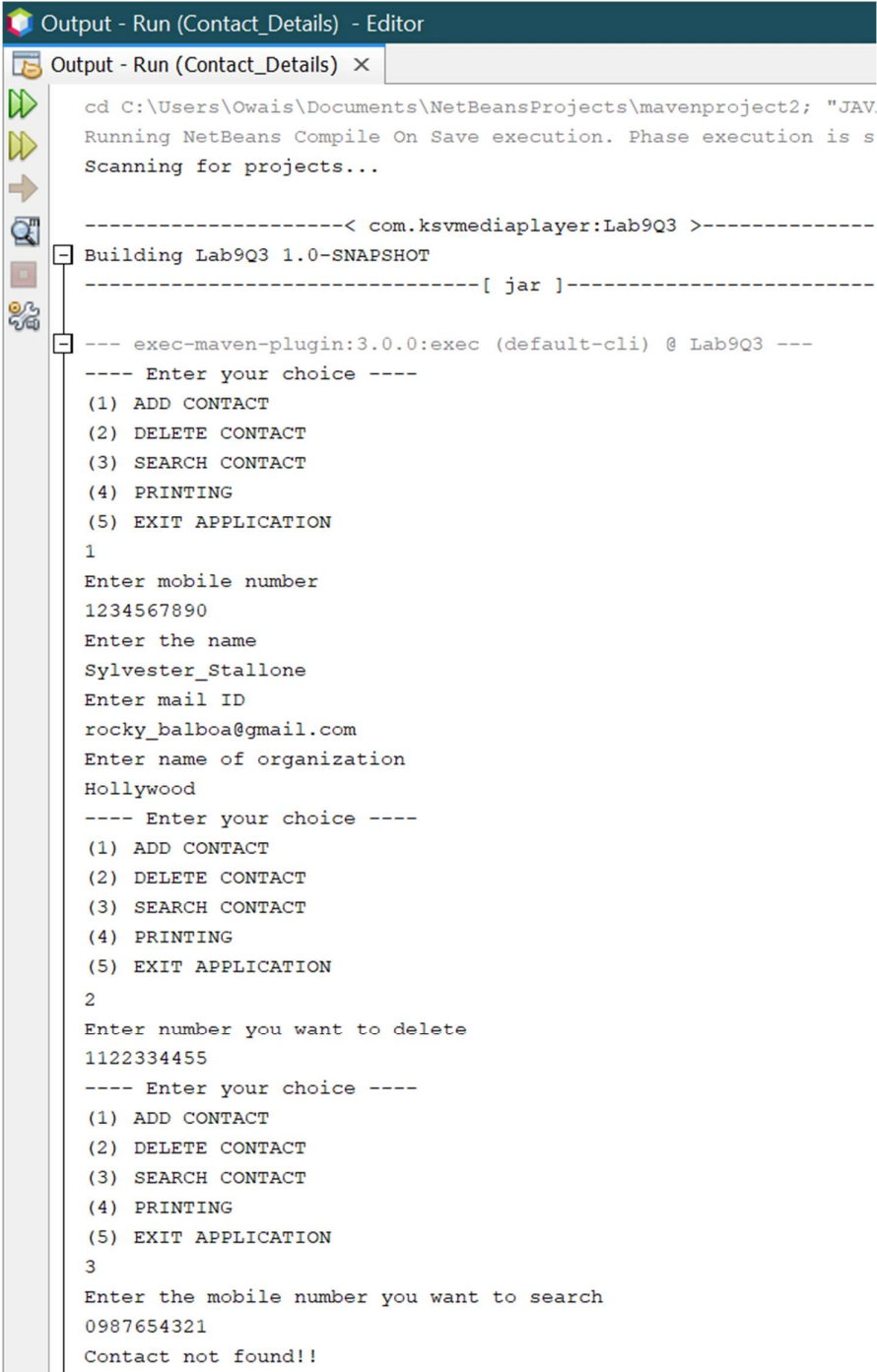
```
Contact_Details.java - Editor
Contact_Details.java ×
Source History
1  /* KHAN MOHD OWAIS RAZA 20BCD7138 */
2  /* CSE2005 LAB-9 */
3  package com.cse2005.lab9Q3;
4  import java.util.*;
5  import java.util.Map.Entry;
6  class info{
7      long mobileNumber;
8      String name;
9      String mails;
10     String organisation;
11     info(long mn,String n,String m,String o){
12         mobileNumber = mn;
13         name=n;
14         mails = m;
15         organisation = o;
16     }
17 }
18 public class Contact_Details{
19     public static void main(String[] args) {
20         HashMap<info,Integer> map = new HashMap<info,Integer>();
21         Scanner sc= new Scanner(System.in);
22         int z=1;
23         boolean kav=true;
24         while(kav){
25             System.out.println("---- Enter your choice ----");
26             System.out.println("(1) ADD CONTACT");
27             System.out.println("(2) DELETE CONTACT");
28             System.out.println("(3) SEARCH CONTACT");
29             System.out.println("(4) PRINTING");
30             System.out.println("(5) EXIT APPLICATION");
31             int pol =sc.nextInt();
32             switch (pol) {
33                 case 1:
34                     System.out.println("Enter mobile number ");
35                     long mn=sc.nextLong();
36                     System.out.println("Enter the name");
37                     String name =sc.next();
38                     System.out.println("Enter mail ID");
39                     String mail =sc.next();
40                     System.out.println("Enter name of organization");
41                     String orga=sc.next();
```

```

42     map.put(new info(mn,name,mail,orga), z);
43     z++;
44     break;
45     case 2:
46         System.out.println("Enter number you want to delete");
47         long eer=sc.nextLong();
48         for (Entry<info, Integer> e : map.entrySet()) {
49             if(e.getKey().mobileNumber==eer){
50                 map.remove(e.getKey());
51                 System.out.println("Deleted contact with name "
52                     +e.getKey().name);
53             }
54         }
55         break;
56     case 3:
57         System.out.println("Enter the mobile number you want to search");
58         long search_mn=sc.nextLong();
59         boolean flag=false;
60         for (Entry<info, Integer> e : map.entrySet()) {
61             if(e.getKey().mobileNumber==search_mn){
62                 System.out.println("Contact found!! at value: "
63                     +e.getValue()
64                     +" name:"+e.getKey().name);
65                 flag=true;
66             }
67         }
68         if(!flag){
69             System.out.println("Contact not found!!");
70         }
71         break;
72     case 4:
73         System.out.println("Printing!!! Mobile number and name");
74         for (Entry<info, Integer> e : map.entrySet()) {
75             System.out.println(e.getKey().mobileNumber+" "
76                 +e.getKey().name);
77         }
78         break;
79     case 5:
80         System.out.println("Exiting");
81         kav=false;
82         break;
83
84     default:
85         break;
86 }
87 }
88 }
89 }
90 }

```


Output console –



```
Output - Run (Contact_Details) - Editor
Output - Run (Contact_Details) X
cd C:\Users\Owais\Documents\NetBeansProjects\mavenproject2; "JAV
Running NetBeans Compile On Save execution. Phase execution is s
Scanning for projects...

-----< com.ksvmediaplayer:Lab9Q3 >-----
Building Lab9Q3 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Lab9Q3 ---
---- Enter your choice ----
(1) ADD CONTACT
(2) DELETE CONTACT
(3) SEARCH CONTACT
(4) PRINTING
(5) EXIT APPLICATION
1
Enter mobile number
1234567890
Enter the name
Sylvester_Stallone
Enter mail ID
rocky_balboa@gmail.com
Enter name of organization
Hollywood
---- Enter your choice ----
(1) ADD CONTACT
(2) DELETE CONTACT
(3) SEARCH CONTACT
(4) PRINTING
(5) EXIT APPLICATION
2
Enter number you want to delete
1122334455
---- Enter your choice ----
(1) ADD CONTACT
(2) DELETE CONTACT
(3) SEARCH CONTACT
(4) PRINTING
(5) EXIT APPLICATION
3
Enter the mobile number you want to search
0987654321
Contact not found!!
```

---- Enter your choice ----

- (1) ADD CONTACT
- (2) DELETE CONTACT
- (3) SEARCH CONTACT
- (4) PRINTING
- (5) EXIT APPLICATION

4

Printing!!! Mobile number and name

1234567890 Sylvester_Stallone

---- Enter your choice ----

- (1) ADD CONTACT
- (2) DELETE CONTACT
- (3) SEARCH CONTACT
- (4) PRINTING
- (5) EXIT APPLICATION

5

Exiting

BUILD SUCCESS

Total time: 02:39 min

Finished at: 2021-12-17T14:10:50+05:30

<