PG DAC–March 2023
C-DAC THIRUVANANTHAPURAM
JAVA- LAB 13

Q1. Write a Java program to iterate through all elements in a array list

IterateArray.java × *RemoveElem.java

```
1  package com.javaassignment131.main;
2  import java.util.Iterator;
3  import java.util.ArrayList;
4
5  public class IterateArray {
6
7      public static void main(String[] args) {
8              ArrayList<Integer> lst = new ArrayList<Integer>();
9
10             lst.add(100);
11             lst.add(200);
12             lst.add(300);
13             lst.add(400);
14             lst.add(500);
15
16             System.out.println("ArrayList elements are as follows : ");
17             lst.forEach(System.out::println);
18
19             System.out.println("Itrating through array elements : ");
20             Iterator<Integer> it = lst.iterator();
21
22             while(it.hasNext()) {
23             System.out.println(it.next());
24         }
25     }
26 }
27
```

Console ×

<terminated> IterateArray [Java Application] D:\Eclipse\eclipse\plugins\o

```
ArrayList elements are as follows :
100
200
300
400
500
Itrating through array elements :
100
200
300
400
500
```

Q2. Write a Java program to remove the third element from a array list

IterateArray.java | RemoveElem.java ×

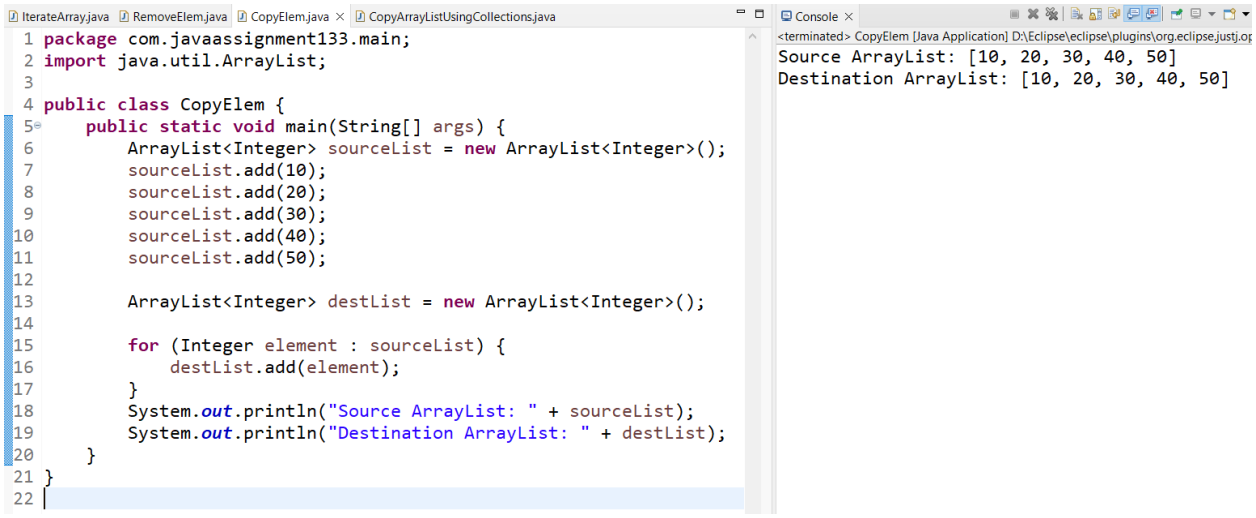
```
1  package com.javaassignment132.main;
2  import java.util.ArrayList;
3  import java.util.Iterator;
4
5  public class RemoveElem {
6
7      public static void main(String[] args) {
8          ArrayList<Integer> lst = new ArrayList<Integer>();
9
10         lst.add(100);
11         lst.add(200);
12         lst.add(300);
13         lst.add(400);
14         lst.add(500);
15
16         System.out.println("List of elements are as follows: ");
17         Iterator<Integer> it = lst.iterator();
18
19         while(it.hasNext()) {
20             System.out.println(it.next());
21         }
22         lst.remove(3);
23         System.out.println("List after removing 3rd element: ");
24         lst.forEach(System.out::println);
25 }
26     }
27
```

Console ×

<terminated> RemoveElem [Java Application] D:\Eclipse\eclipse\plugi

```
List of elements are as follows:
100
200
300
400
500
List after removing 3rd element:
100
200
300
500
```

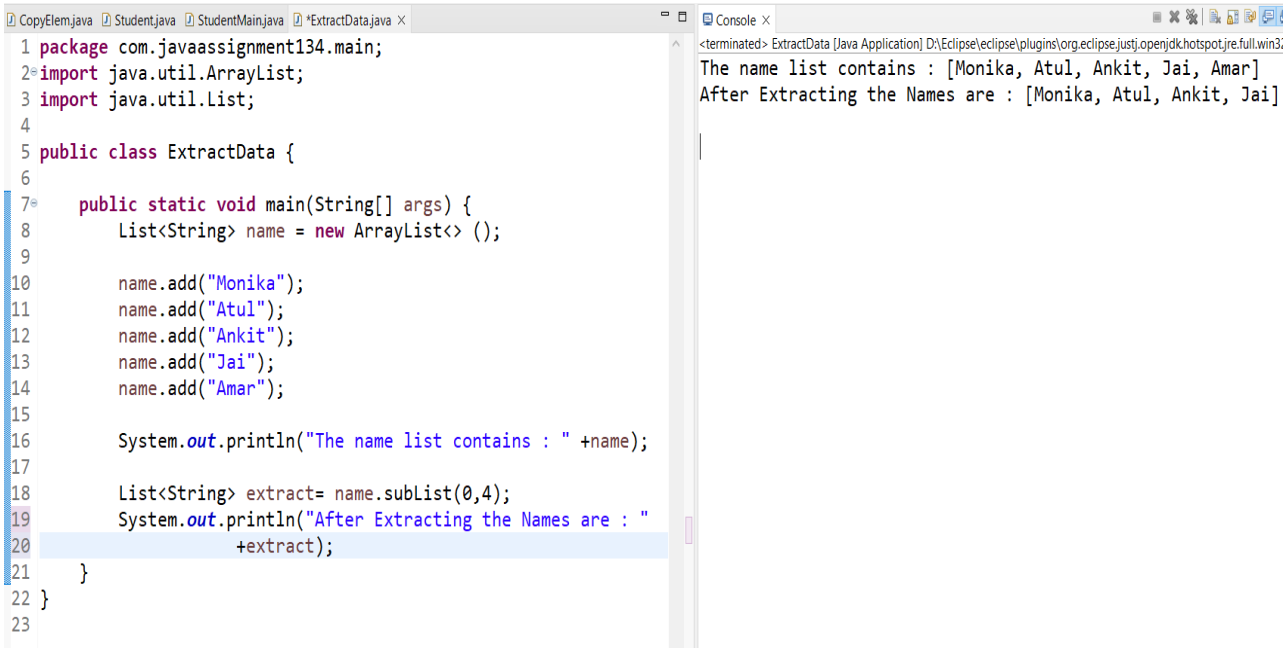
Q3. Write a Java program to copy one array list into another.



```java
package com.javaassignment133.main;
import java.util.ArrayList;

public class CopyElem {
    public static void main(String[] args) {
        ArrayList<Integer> sourceList = new ArrayList<Integer>();
        sourceList.add(10);
        sourceList.add(20);
        sourceList.add(30);
        sourceList.add(40);
        sourceList.add(50);

        ArrayList<Integer> destList = new ArrayList<Integer>();

        for (Integer element : sourceList) {
            destList.add(element);
        }
        System.out.println("Source ArrayList: " + sourceList);
        System.out.println("Destination ArrayList: " + destList);
    }
}
```

```
<terminated> CopyElem [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.o
Source ArrayList: [10, 20, 30, 40, 50]
Destination ArrayList: [10, 20, 30, 40, 50]
```

Q4. Write a Java program to extract a portion of a array list.



```java
package com.javaassignment134.main;
import java.util.ArrayList;
import java.util.List;

public class ExtractData {

    public static void main(String[] args) {
        List<String> name = new ArrayList<> ();

        name.add("Monika");
        name.add("Atul");
        name.add("Ankit");
        name.add("Jai");
        name.add("Amar");

        System.out.println("The name list contains : " +name);

        List<String> extract= name.subList(0,4);
        System.out.println("After Extracting the Names are : "
                +extract);
    }
}
```

```
<terminated> ExtractData [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win3
The name list contains : [Monika, Atul, Ankit, Jai, Amar]
After Extracting the Names are : [Monika, Atul, Ankit, Jai]
```

Q5. Create a class named Student with following Data members - regno, name, marks. Create 2 array lists- list1 & list2 - of type Student. Create 4 objects of Student class. Add 2 objects of Student class and add to list1 and remaining 2 objects to list2. Compare list1 & list2 to find whether both have same data

CopyElem.java | Student.java | StudentMain.java

```
1  package com.javaassignment135.main;
2  import java.util.Objects;
3
4  public class Student {
5      private String name;
6      private int regno;
7      private int marks;
8  public Student(String name, int regno, int marks){
9      this.name = name;
10     this.regno = regno;
11     this.marks = marks;
12 }
13 public Student() {
14 }
15 public String getName() {
16     return name;
17 }
18 public void setName(String name) {
19     this.name = name;
20 }
21 public int getRegno() {
22     return regno;
23 }
24 public void setRegno(int regno) {
25     this.regno = regno;
26 }
27 public int getMarks() {
28     return marks;
29 }
30 public void setMarks(int marks) {
31     this.marks = marks;
32 }
```

```
33 @Override
34 public String toString() {
35 return "Reg No: " + regno + ", Name: " + name + ", Marks: " + marks;
36 }
37 @Override
38 public int hashCode() {
39     return Objects.hash(marks, name, regno);
40 }
41 @Override
42 public boolean equals(Object obj) {
43     if (this == obj)
44         return true;
45     if (obj == null)
46         return false;
47     if (getClass() != obj.getClass())
48         return false;
49     Student other = (Student) obj;
50     return marks == other.marks && Objects.equals(name, other.name) && regno == other.regno;
51 }
52
53 }
54
```

CopyElem.java | Student.java | StudentMain.java

```java
package com.javaassignment135.main;
import java.util.ArrayList;

public class StudentMain {

    public static void main(String[] args) {
        ArrayList<Student> list1 = new ArrayList<Student>();
        ArrayList<Student> list2 = new ArrayList<Student>();

        Student s1 = new Student("Monika", 101, 90);
        Student s2 = new Student("Ankit", 102, 80);
        Student s3 = new Student("Ankit", 102, 80);
        Student s4 = new Student("Monika", 101, 90);

        System.out.println("List1 objects are: ");
        list1.add(s1);
        list1.add(s2);
        list1.forEach(System.out::println);

        System.out.println("List2 objects are: ");
        list2.add(s3);
        list2.add(s4);
        list2.forEach(System.out::println);

        if(list1.containsAll(list2) && list2.containsAll(list1))
        System.out.println("Both list are same");
        else
            System.out.println("Both list are different");
    }
}
```

Console

```
<terminated> StudentMain (1) [Java Application] D:\Eclipse\eclipse\plugins\c
List1 objects are:
Reg No: 101, Name: Monika, Marks: 90
Reg No: 102, Name: Ankit, Marks: 80
List2 objects are:
Reg No: 102, Name: Ankit, Marks: 80
Reg No: 101, Name: Monika, Marks: 90
Both list are same
```

Q6. Write a Java program to create a Linked List of Integers and insert the specified element at the specified position in the linked list.

CopyElem.java | Student.java | StudentMain.java | ExtractData.java | *LinkedListMain.java

```java
package com.javaassignment136.main;

import java.util.LinkedList;

public class LinkedListMain {

    public static void main(String[] args) {
        LinkedList<Integer> lst = new LinkedList<Integer>();

        lst.add(200);
        lst.add(300);
        lst.add(400);
        lst.add(500);
        System.out.println("All elements from list are:");
        lst.forEach(System.out::println);

        lst.addFirst(100);
        lst.addLast(600);

        System.out.println("All elements from list after "
                + "inserting the specified elements:");
        lst.forEach(System.out::println);
    }
}
```

Console

```
<terminated> LinkedListMain [Java Application] D:\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hot
All elements from list are:
200
300
400
500
All elements from list after inserting the specified
100
200
300
400
500
600
```

Q7. Write a Java program to iterate a linked list.

Student.java | StudentMain.java | IterateLinklist.java | ReverseIterator.java

```
1  package com.javaassignment137.main;
2  import java.util.Iterator;
3  import java.util.LinkedList;
4  public class IterateLinklist {
5
6      public static void main(String[] args) {
7          LinkedList<Integer> lst = new LinkedList<Integer>();
8
9          lst.add(20);
10         lst.add(30);
11         lst.add(40);
12         lst.add(50);
13         lst.add(60);
14
15     System.out.println("Iterating through array elements : ");
16     Iterator<Integer> it = lst.iterator();
17
18     while(it.hasNext()) {
19     System.out.println(it.next());
20         }
21     }
22 }
23
24
```

Console

```
<terminated> IterateLinklist [Java Application] D:\Eclipse\eclipse\plu
Iterating through array elements :
20
30
40
50
60
```

Q8. Write a Java program to iterate a linked list in reverse order.

Student.java | StudentMain.java | IterateLinklist.java | *ReverseIterator.java

```
1  package com.javaassignment138.main;
2
3  import java.util.Iterator;
4  import java.util.LinkedList;
5
6  public class ReverseIterator {
7
8      public static void main(String[] args) {
9          LinkedList<Integer> lst = new LinkedList<Integer>();
10
11         lst.add(10);
12         lst.add(20);
13         lst.add(30);
14         lst.add(40);
15         lst.add(50);
16
17         System.out.println("Oiginal list of elements : ");
18         lst.forEach(System.out::println);
19
20         System.out.println("Reverse list of elements : ");
21         Iterator<Integer> descendingIterator = lst.descendingIterator();
22         while(descendingIterator.hasNext()) {
23         System.out.println(descendingIterator.next());
24     }
25 }
26     }
27
```

Console

```
<terminated> ReverseIterator [Java Application] D:\Eclipse
Oiginal list of elements :
10
20
30
40
50
Reverse list of elements :
50
40
30
20
10
```

Q9. Create a class named Student with following Data members - regno, name, marks. Create a HashSet of type Student. Create Student objects and store in the HashSet in a way that One Student's data (data with same roll no and name) can be stored only once in the HashSet.

Student.java × StudentMain.java *Student.java

```java
package com.javaassignment139.main;

import java.util.Objects;

public class Student {
        private int regno;
        private String name;
        private int marks;
    public Student(int regno, String name, int marks){
        this.regno = regno;
        this.name = name;
        this.marks = marks;
    }
    public Student() {
    }
    public int getRegno() {
        return regno;
    }
    public void setRegno(int regno) {
        this.regno = regno;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getMarks() {
        return marks;
    }
    public void setMarks(int marks) {
        this.marks = marks;
    }
    @Override
    public String toString() {
    return "Reg No: " + regno + ", Name: " + name + ", Marks: " + marks;
    }
    @Override
    public int hashCode() {
        return Objects.hash(marks, name, regno);
    }
    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Student other = (Student) obj;
        return marks == other.marks && Objects.equals(name, other.name) && regno == other.regno;
    }
}

```

Student.java | StudentMain.java | ReverseIterator.java | Student.java | StudentMain.java

```java
package com.javaassignment139.main;
import java.util.HashSet;


public class StudentMain {

    public static void main(String[] args) {
        HashSet<Student> hset = new HashSet<Student>();
        Student s1 = new Student(101, "Monika", 90);
        Student s2 = new Student(101, "Monika", 90);
        Student s3 = new Student(102, "Ankit", 85);
        Student s4 = new Student(103, "Amar", 80);
        Student s5 = new Student(104, "Kunal", 90);

        hset.add(s5);
        hset.add(s4);
        hset.add(s3);
        hset.add(s2);
        hset.add(s1);

        System.out.println("All students from hset are: ");
        hset.forEach(System.out::println);

    }
}

```

Console

```
<terminated> StudentMain (2) [Java Application] D:\Eclipse\eclipse\plugi
All students from hset are:
Reg No: 103, Name: Amar, Marks: 80
Reg No: 101, Name: Monika, Marks: 90
Reg No: 104, Name: Kunal, Marks: 90
Reg No: 102, Name: Ankit, Marks: 85
```

Q10. Create a class named Student with following Data members - regno, name, marks. Create a TreeSet of type Student. Create Student objects and store in the TreeTet .

Student.java | StudentMain.java | Student.java | StudentMain.java

```java
package com.javaassignment1310.main;
import java.util.Objects;

public class Student implements Comparable<Student>{
        private int regno;
        private String name;
        private int marks;
    public Student(int regno, String name, int marks){
        this.regno = regno;
        this.name = name;
        this.marks = marks;
    }
    public Student() {
    }
    public int getRegno() {
        return regno;
    }
    public void setRegno(int regno) {
        this.regno = regno;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getMarks() {
        return marks;
    }
    public void setMarks(int marks) {
        this.marks = marks;
    }
```

```
    @Override
    public String toString() {
    return "Reg No: " + regno + ", Name: " + name + ", Marks: " + marks;
    }
@Override
    public int hashCode() {
        return Objects.hash(marks, name, regno);
    }
    @Override
    public boolean equals(Object obj) {
    if (this == obj)
        return true;
        if (obj == null)
            return false;
        if (getClass() != obj.getClass())
            return false;
        Student other = (Student) obj;
    return marks == other.marks && Objects.equals(name, other.name) && regno == other.regno;
    }
public int compareTo(Student obj) {
    if(this.regno ==obj.regno)
        return 0;
    else
        return -1;
    }
}

```

Student.java | StudentMain.java | Student.java | StudentMain.java

```
package com.javaassignment1310.main;

import java.util.TreeSet;

public class StudentMain {

    public static void main(String[] args) {
        TreeSet<Student> eset = new TreeSet<Student>();
        Student s1 = new Student(101, "Monika", 90);
        Student s2 = new Student(102, "Ankit", 85);
        Student s3 = new Student(103, "Amar", 80);
        Student s4 = new Student(104, "Kunal", 90);

        eset.add(s1);
        eset.add(s2);
        eset.add(s3);
        eset.add(s4);

        System.out.println("All students from TreeSet are: ");
        eset.forEach(System.out::println);
    }
}

```

Console

```
<terminated> StudentMain (3) [Java Application] D:\Eclipse\eclipse\plug
All students from TreeSet are:
Reg No: 104, Name: Kunal, Marks: 90
Reg No: 103, Name: Amar, Marks: 80
Reg No: 102, Name: Ankit, Marks: 85
Reg No: 101, Name: Monika, Marks: 90
```

Q11. Write a Java program to store key, value pairs in a Map.

Student.java | StudentMain.java | Student.java | StudentMain.java | MapMain.java

```
1  package com.javaassignment1311.main;
2
3  import java.util.Map;
4  import java.util.Map.Entry;
5  import java.util.Set;
6  import java.util.HashMap;
7
8  public class MapMain {
9
10     public static void main(String[] args) {
11         Map<Integer, String> hmap = new HashMap<>();
12
13         hmap.put(1234, "abcd");
14         hmap.put(5678, "efgh");
15         hmap.put(4321, "ijkl");
16         hmap.put(8765, "mnop");
17
18         System.out.println("All enteries from hmap are: ");
19         Set<Entry<Integer, String>> eset = hmap.entrySet();
20
21         for(Entry<Integer, String> e : eset)
22             System.out.println(e.getKey() +" : " + e.getValue());
23
24     }
25 }
26
```

Console

```
<terminated> MapMain [Java Application] D:\Eclipse\ecli
All enteries from hmap are:
4321 : ijkl
1234 : abcd
8765 : mnop
5678 : efgh
```