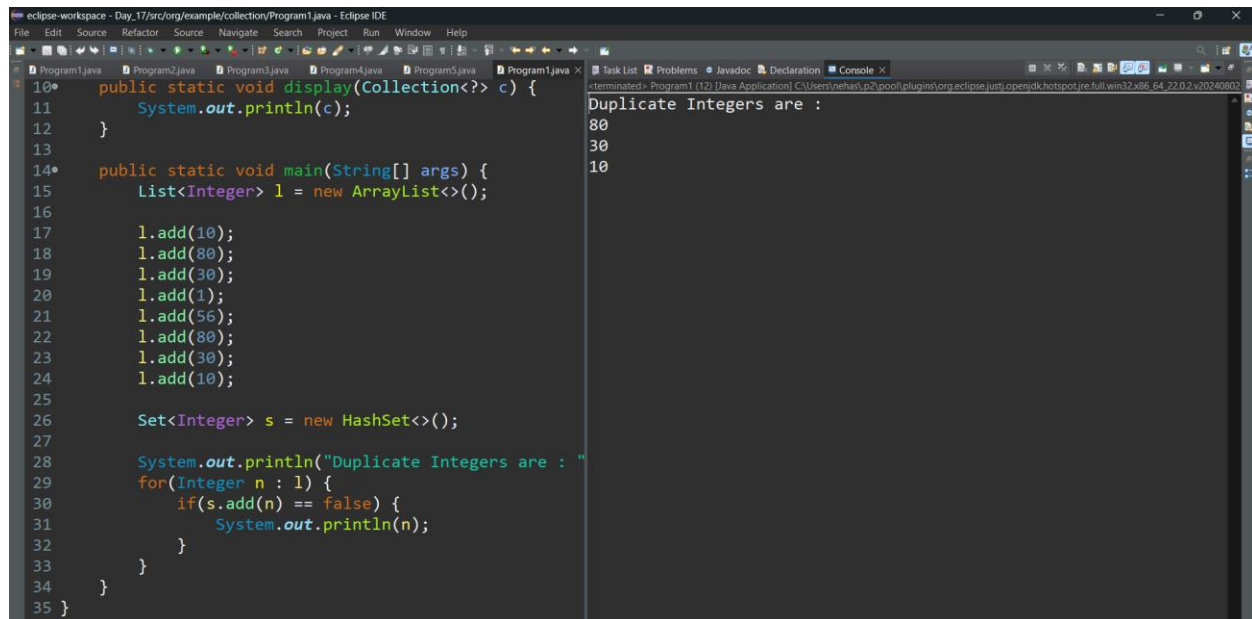


PG-DAC AUGUST 24 BATCH

1) Write a Java program that takes a list of integers as input and returns a list of duplicate integers.

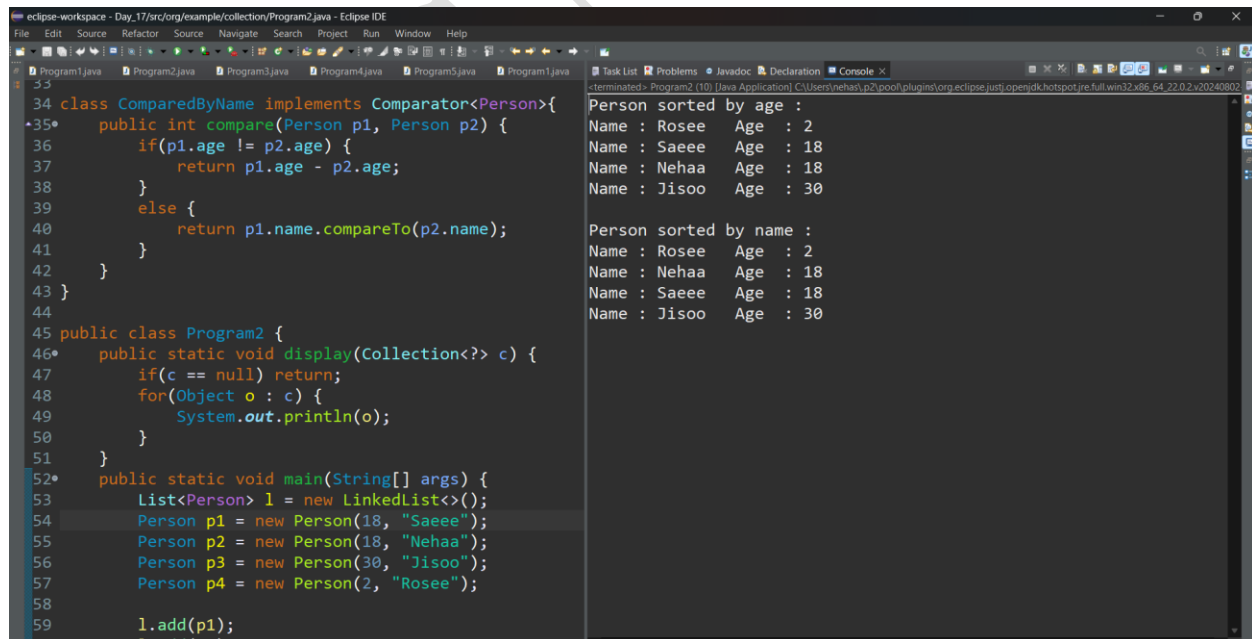


```
10 public static void display(Collection<?> c) {
11     System.out.println(c);
12 }
13
14 public static void main(String[] args) {
15     List<Integer> l = new ArrayList<>();
16
17     l.add(10);
18     l.add(80);
19     l.add(30);
20     l.add(1);
21     l.add(56);
22     l.add(80);
23     l.add(30);
24     l.add(10);
25
26     Set<Integer> s = new HashSet<>();
27
28     System.out.println("Duplicate Integers are : ");
29     for(Integer n : l) {
30         if(s.add(n) == false) {
31             System.out.println(n);
32         }
33     }
34 }
35 }
```

Console Output:

```
Duplicate Integers are :
80
30
10
```

2) Create a Person class with attributes name and age. Write a Java program that sorts a list of Person objects first by age and then by name if the ages are equal.



```
34 class ComparedByName implements Comparator<Person>{
35     public int compare(Person p1, Person p2) {
36         if(p1.age != p2.age) {
37             return p1.age - p2.age;
38         }
39         else {
40             return p1.name.compareTo(p2.name);
41         }
42     }
43 }
44
45 public class Program2 {
46     public static void display(Collection<?> c) {
47         if(c == null) return;
48         for(Object o : c) {
49             System.out.println(o);
50         }
51     }
52
53     public static void main(String[] args) {
54         List<Person> l = new LinkedList<>();
55         Person p1 = new Person(18, "Saeed");
56         Person p2 = new Person(18, "Nehaa");
57         Person p3 = new Person(30, "Jisoo");
58         Person p4 = new Person(2, "Rosee");
59
60         l.add(p1);
61
62         display(l);
63     }
64 }
```

Console Output:

```
Person sorted by age :
Name : Rosee Age : 2
Name : Saeed Age : 18
Name : Nehaa Age : 18
Name : Jisoo Age : 30

Person sorted by name :
Name : Rosee Age : 2
Name : Nehaa Age : 18
Name : Saeed Age : 18
Name : Jisoo Age : 30
```

3) Write a Java program to find the first non-repeated character in a string using a HashMap. String input = "aabbccddeffg"; Expected output = 'e'

```
eclipse-workspace - Day_17/src/org/example/collection/Program4.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Program1.java Program2.java Program3.java Program4.java
1 package org.example.collection;
2
3 import java.util.Map;
4 import java.util.LinkedHashMap;
5
6 public class Program4 {
7     public static void main(String[] args) {
8         String input = "aabbccddeffg";
9         System.out.print("First non repeat character in a String is : ");
10        Map<Character, Integer> m = new LinkedHashMap<>();
11
12        for(Character c : input.toCharArray()) {
13            m.put(c, m.get(c) == null ? 1 : m.get(c) + 1);
14        }
15
16        for(Character c : m.keySet()) {
17            if(m.get(c) == 1) {
18                System.out.println(c);
19                break;
20            }
21        }
22    }
23 }
24
First non repeat character in a String is : e
<terminated> Program4 (3) [Java Application] C:\Users\nehas.p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.22.0.2\jre\bin\java.exe
```

4) Write a Java program that merges two sorted lists of integers into a single sorted list.

```
eclipse-workspace - Day_17/src/org/example/collection/Program3.java - Eclipse IDE
File Edit Source Refactor Source Navigate Search Project Run Window Help
Program1.java Program2.java Program3.java Program4.java
1 package org.example.collection;
2
3 import java.util.List;
4 import java.util.Collection;
5 import java.util.Collections;
6 import java.util.LinkedList;
7
8 public class Program3 {
9     public static void display(Collection<?> c) {
10        if(c == null) return;
11        for(Object o : c) {
12            System.out.println(o);
13        }
14    }
15
16     public static void main(String[] args) {
17        List<Integer> l1 = new LinkedList<>();
18        List<Integer> l2 = new LinkedList<>();
19        List<Integer> l3 = new LinkedList<>();
20
21        l1.add(10);
22        l1.add(50);
23        l1.add(20);
24
25        l2.add(30);
26
27        Sorted List 1 :
28        10
29        20
30        50
31        Sorted List 2 :
32        30
33        40
34        60
35        80
36        Sorted List 3 :
37        70
38        90
39        100
40        List 1 & 2 added to List 3 :
41        70
42        90
43        100
44        10
45        20
46        50
47        30
48        40
49        60
50        80
51    }
52 }
```