

ArrayList, Abstract Class and Interface

Lab Objective

Familiarize students with the implementation of abstract class and interface in Java.

Lab Outcomes

After completing this lab successfully, students will be able to:

1. **Write** the definition of an abstract class.
2. **Write** codes to implement interfaces.

Psychomotor Learning Levels

This lab involves activities that encompass the following learning levels in psychomotor domain.

Level	Category	Meaning	Keywords
P1	Imitation	Copy action of another; observe and replicate.	Relate, Repeat, Choose, Copy, Follow, Show, Identify, Isolate.
P2	Manipulation	Reproduce activity from instruction or memory	Copy, response, trace, Show, Start, Perform, Execute, Recreate.

Lab Problems

A. Comparator Interface

Comparators are used to compare two objects. In this problem, you'll create a comparator and use it to sort an array.

The *Player* class is provided for you in your editor. It has 2 fields: a *String* and a *integer*.

Given an array of *n Player* objects, write a comparator that sorts them in order of decreasing score; if 2 or more players have the same score, sort those players alphabetically by name. To do this, you must create a *Checker* class that implements the *Comparator* interface, then write an *int compare(Player a, Player b)* method implementing the *Comparator.compare(T o1, T o2)* method.

Input Format

The first line contains an integer, *n*, denoting the number of players.

Each of the *n* subsequent lines contains a player's *name* and *score*, respectively.

Output Format

Print each sorted element in the format: *name score*

Sample Input

```
5
amy 100
david 100
heraldo 50
aakansha 75
aleksa 150
```

Sample Output

```
aleksa 150
amy 100
david 100
aakansha 75
heraldo 50
```

Problem B. Java ArrayList

Sometimes it's better to use dynamic size arrays. Java's ArrayList can provide you this feature. Try to solve this problem using ArrayList.

You are given n lines. In each line there are zero or more integers. You need to answer a few queries where you need to tell the number located in y^{th} position of x^{th} line. Take your input from System.in.

Input Format

The first line has an integer n . In each of the next n lines there will be an integer d denoting number of integers on that line and then there will be d space-separated integers. In the next line there will be an integer q denoting number of queries. Each query will consist of two integers x and y .

Output Format

In each line, output the number located in y^{th} position of x^{th} line. If there is no such position, just print "ERROR!"

Sample Input

```
5
5 41 77 74 22 44
1 12
4 37 34 36 52
0
3 20 22 33
5
1 3
3 4
3 1
4 3
5 5
```

Sample Output

```
74
52
37
ERROR!
ERROR!
```