JAVA Practice problems (part-1)

These are practice problems only to reinforce concepts discussed so far

You MUST not consider them as representative question for exams

---

1. Write **COMPLETE java program** (Test.java) for the following:

```
class Complex {
    int real, img;
    //add constructor and other necessary methods for input, output
} //represents complex number in the form of "real +/- img i" e.g: 2+3i

class ComplexArray {
    Complex[ ] data;
    //write populate & show methods
}

//rewrite TEST class with full functioning code
public class Test {
  public static void main(){
    int n;
    //get values of n from user (no of Complex numbers)
    ComplexArray arr;
    //allocate necessary memory to arr
    Loop: as long as user wants [choice: 1 to populate, 2 to exit]
    {
       arr.augment(n); // add/augment n complex numbers to arr
    }
    int lower, upper;
    //get values of lower & upper from user

    Now display all the Complex numbers from arr whose real<=lower &
    img>=upper, using show(lower,upper) method of Complex class;
   }
}
```

P.T.O

2. **Complete the following incomplete code to a Full java program (Test.java):**

```
class MyArray {
   int[] intData;
   // add necessary methods
}

class Matrix {
   MyArray[ ][ ] arrayObjects;
   // add necessary methods ( see main() )
}

public class Test {
  public static void main(){
   Matrix m = new Matrix();
   Loop: as long as user wants to continue
   {
      m.augment();
      // get values of # of rows & cols from user.
      // if m is empty, instantiate the 2D array. Now ask size of each
      // MyArray objects of the 2D and populate them with random integers.

      // if m already has some MyArray objects in it, and IFF rows &
      // cols are greater than the row & column no of m, then ask size
      // of each additional MyArray objects of the 2D and augment them
      // to m, populated with user input.


      m.display();
      // display the Matrix of MyArray objects
   }
  }//end main()
}//end class
```

P.T.O

3. Create a netbeans java project with two packages: myarrays and mainpkg

- Classes of your projects are: myarrays.***OneDArray***, myarrays.**Matrix** and mainpkg.**MainClass**
- Class OneDArray has following **private** fields: **int[] values, float average**
  - Methods: a) **void getArray();** b) **void showArray();**
  - If necessary, add other methods to ensure that your main method works
- Matrix class has following **private** fields: **OneDArray[][] arrays**
  - According to given RUN, you need to add appropriate methods in Matrix class

MainClass has the following main method:

```
public static void main(String[] args){
    Matrix m1, m2, m3;
    r = no of rows for Matrix class object. r is a user input
    c = no of columns for Matrix class object. c is a user input

    m1 = new Matrix(r, c); // m1 will have r rows & c cols
      //stores OneDArray objects in arrays[i][j] inside m1 object
      //ask user for length and values for each OneDArray
      //average value of each MyOneDArray is also calculated

      Sout(“First Matrix:”); m1.showMatrix(); //see RUN

    m2 = new Matrix(r, c, 2, 10); // m2 will have r rows & c cols
      //3rd parameter is the length of first OneDArray in m2,
      //which gets incremented by 1 for subsequent OneDArray objects in m2
      //4th parameter is the upper limit of random values to populate m2
      //average value of each OneDArray is also calculated

    Sout(“Second Matrix:”); m2.showMatrix();//see RUN

    m3 = m1.merge(m2);

    Sout(“Merged Matrix:”); m3.showMatrix();//see RUN
}
```

RUN:

```
How many rows? 2
How many columns? 2

How many numbers: 2
Enter values: 1 3
How many numbers: 3
Enter values: 4 8 6
How many numbers: 2
Enter values: 7 2
How many numbers: 4
Enter values: 3 7 9 1
```

```
First Matrix:
{1,3} Avg: 2                {4,8,6} Avg: 6
{7,2} Avg: 4.5              {3,7,9,1}  Avg: 5

Second Matrix:
{5,1}  Avg: 3                {3,1,7} Avg: 3.67
{9,1,5,4} Avg: 4.75          {2,7,1,8,5} Avg: 4.6

Merged Matrix:
{1,3,5,1} Avg: 2.5           {4,8,6,3,1,7} Avg: 4.83
{7,2,9,1,5,4} Avg: 4.67      {3,7,9,1,2,7,1,8,5} Avg: 4.78
```

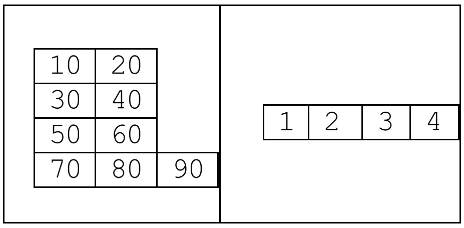
Now,

- Implement the above project without changing main(), fields of the classes and given RUN

P.T.O

4. Write **COMPLETE java program** (Test.java) for the following:

Declare and populate a 3D Java array called **arr3** as per the following diagram:



10	20	
30	40	
50	60	
70	80	90

1	2	3	4

P.T.O

5. **Create a netbeans java project to implement the following**

Your console application package called **mypkg** has the following:

- class **Student** with fields (id, name, cgpa, dept, major), constructors, setter & getter methods
- public class **MainClass** having:
  - private field: an ArrayList of **Student** objects called **studArray**
  - private field: an array of ArrayList<float> **cgpaTable**
  - Private method **populateStudArray** which reads Student information from user for n students (n is a user input and ensure that n is >0 and <=45) and stores them in studArray. Now, if same ID is given by user for two students, you must not proceed (show custom message) until user gives a unique ID.
  - private method **scanStudArray** to go through the already loaded **studArray** and copy student cgpa values into an array of 3 ArrayList (s) of floats named **cgpaTable [SIMULATING 2D ARRAY]**, where
    - all the cgpa (s) < 2.0 to be added to cgpaTable[0]
    - all the cgpa (s) >= 2.0 and <3.0 to be added to cgpaTable[1]
    - all the cgpa (s) >= 3.0 to be added to cgpaTable[2]
  - private method **displayCgpaTable** to print 3 rows of cgpas from **cgpaTable** to the console separated by comma
  - public static void main(…){

    //menu based do-while loop to call **populateStudArray, scanStudArray and displayCgpaTable**

    }

WRITE Student class with appropriate fields & methods
WRITE MainClass class with appropriate field declaration
WRITE populateStudArray method of MainClass class
WRITE scanStudArray method of MainClass class
WRITE displayCgpaTable method of MainClass class
WRITE main method of MainClass class

P.T.O

6. Solve the netbeans project on ARRAY that was discussed as homework:

```
Exercise:

class Student{
   private fields: id, name, cgpa
   private String[][] grades;
   private String[][] credits;
   private void calcCgpa(){//...}
   public void setStudInfo(){
        //after taking id, name, all grades
        calcCgpa();
   }
   static public float getAvgCgpa(Student[] s){
    //...
    //s[i].cgpa, could be ok if getAvgCgpa was NON-static
    total += s[i].getCgpa();
    return total/s.length;
   }
}

psvm(){
    Student[] studArr;
    n = s.nextInt(); //n is no of students
    studArr = new ...

    loop: studArr[i].setStudInfo();
    sout("Average Cgpa = "+ Student.getAvgCgpa(studArr));
}
```