

Solution 1: An array per student

```
double [] student1 = new double[10];  
...  
double [] student8 = new double[10];
```

student1:

7.5	4.5	9.5	2	7	8.5	1	8.5	7.5	6
-----	-----	-----	---	---	-----	---	-----	-----	---

...

student8:

4.5	2	9	9	5.5	4	7.5	6	5	9
-----	---	---	---	-----	---	-----	---	---	---

Tedious!

3

Solution 2: An array per quiz

```
double [] quiz1 = new double[8];  
...  
double [] quiz10 = new double[8];
```

quiz1:

7.5	4.5
-----	-----	-----	-----	-----	-----	-----	-----

...

quiz10:

6	9
---	-----	-----	-----	-----	-----	-----	---

Tedious!

4

Multi-dimensional arrays

How can we store a table of values?

Let's say there's a class of 8 students,
with 10 quiz grades each:

7.5	4.5	9.5	2	7	8.5	1	8.5	7.5	6
...
...
...
...
...
...
4.5	2	9	9	5.5	4	7.5	6	5	9

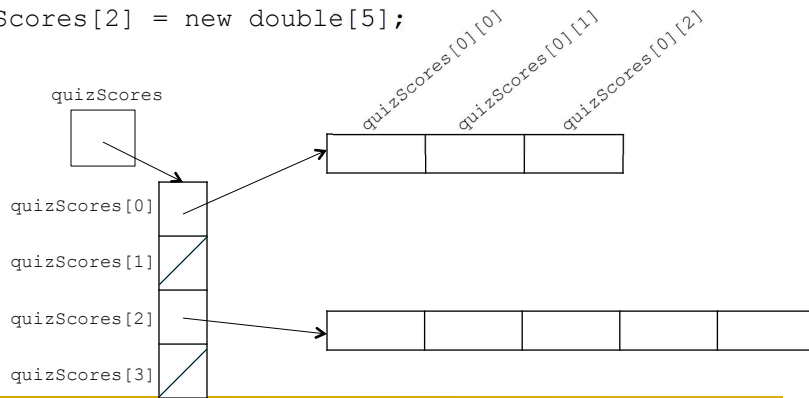
How can we declare variables to store all this data?

1

2

Ragged 2-D arrays

```
double [][] quizScores = new double[4][];  
quizScores[0] = new double[3];  
quizScores[2] = new double[5];
```



Exercises

- Compute the *trace*, or sum of the elements on the diagonal of a matrix.
- Swap each element in a square 2-D array with elements on the opposite side of the main diagonal.

Solution 3: 2-dimensional arrays!

```
double [][] quizScores = new double[8][10];
```

quizScores:

7.5	4.5	9.5	2	7	8.5	1	8.5	7.5	6
...
...
...
...
...
4.5	2	9	9	5.5	4	7.5	6	5	9

quizScores[7][9]

2-D Arrays in memory

- A 2-D array is really an array of arrays!

```
double [][] quizScores = new double[4][3];
```

