COSC 111:
Computer Programming I

Dr. Bowen Hui

University of British Columbia
Okanagan

Review Arrays

- Used to organize sequence of same type of info
- More involved for declaration
  - Template:
    type[] var = new type[N];
- Must initialize separately
  - If not: arrays of objects will be null
- Array of size N has indices 0 to N-1 (inclusive)
- Access i^{th} element of an array via index, e.g. var[i]

Exercise PP8.1 (Pg. 435 of Text)

(Slightly modified)

- Design and implement an application that generates 20 integers that are in the range 0 to 5 inclusive and counts how many occurrences of each are generated.
  - Frequency of numbers in 0 to 5 (like histogram)
- After all the numbers have been processed, print all of the values (with the number of occurrences) that were generated one or more times.

Search Example

- Suppose we have an array of Books and want to know availability for borrowing

```
Book[] library = new Book[ 2000 ];
. . .
for( int i=0; i<library.length; i++ )
    if( library[i].getTitle().equals( targetTitle ) )
    {
        foundindex = i;
        break;              // stop searching once found
    }
```

break

- Reserved word that breaks out of the immediate enclosing loop
- Example:

```
int i;
for( i=0; i<10; i++ )
      if( i % 2 == 0 )
            break;
int y = i;
```

- What is the value of y?

Another Way for Input Validation

- Previously, you read input then checked:

```
while( input < 1 || input > 2 )
{
        System.out.println( “Type 1 or 2 only” );
        input = sysin.nextInt();
}
```

Another Way for Input Validation

- Previously, you read input then checked:

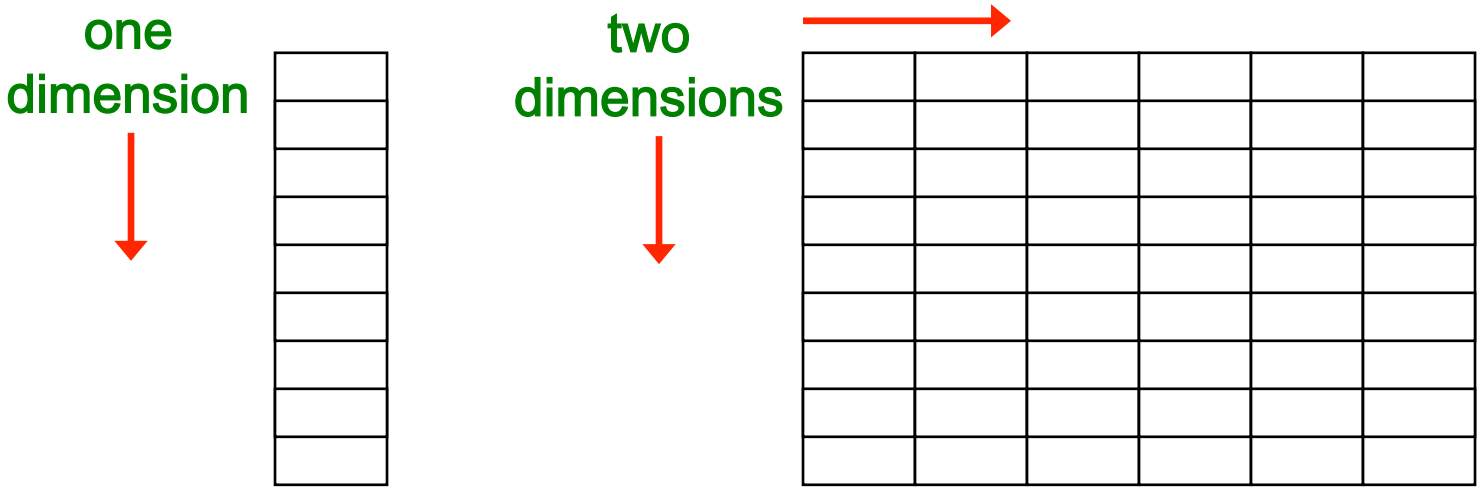
```
while( input < 1 || input > 2 )
{
    System.out.println( “Type 1 or 2 only” );
    input = sysin.nextInt();
}
```

- Equally valid, you read input then now check:

```
while( true )
    if( input == 1 || input == 2 )
        break;
    else
    {
        System.out.println( “Type 1 or 2 only” );
        input = sysin.nextInt();
    }
```

2D Arrays

- A one-dimensional array stores a sequence of elements
- A two-dimensional (2D) array stores a table of elements



Example

- Example:

```
int[][] table = new int[ 12 ][ 50 ];
int row = 3;
int col = 6;
int value = table[ row ][ col ];
int[] lastRow = table[11];
// no easy way to fetch column
```

- This representation, declaration, initialization, use can be generalized to 3D, 4D, etc.
- E.g.: double[][][][][] potential;
  a 5-dimensional array

ClassGrades Example

```
import java.util.Random;

public class ClassGrades
{
  private int[][] allGrades;
  private int numStudents;
  private int numExams;

  public ClassGrades( int numStuds, int numEx )
  {
    numStudents = numStuds;
    numExams = numEx;
    allGrades = new int[ numStudents ][ numExams ];
    for( int e=0; e<numExams; e++ )
      genScores( e );
  }

  public String toString()
  {
    String str = "";
    str += "numStudents = " + numStudents + "\n";
    str += "numExams = " + numExams + "\n";
    str += "allGrades.length = " + allGrades.length + "\n";
    str += "allGrades[0].length = " + allGrades[0].length + "\n";
    return str;
  }
}
```

Running it

- Suppose we called it in the test class:
  ClassGrades cg = new ClassGrades(100, 2);
  System.out.println(cg.toString());
  ClassGrades cg2 = new ClassGrades(5, 10);
  System.out.println(cg2.toString());

Running it

- Suppose we called it in the test class:
  ```
  ClassGrades cg = new ClassGrades( 100, 2 );
  System.out.println( cg.toString() );
  ClassGrades cg2 = new ClassGrades( 5, 10 );
  System.out.println( cg2.toString() );
  ```
- Output:
  ```
  numStudents = 100
  numExams = 2
  allGrades.length = 100
  allGrades[0].length = 2

  numStudents = 5
  numExams = 10
  allGrades.length = 2
  allGrades[0].length = 10
  ```

Completing ClassGrades

- Write a method `genScores()` that generates random exam scores for the i'th exam
- Write a method called `getOneStudent()` that gets all the grades for the i'th student
  - Same as getting a row in a 2D array
- Write a method called `getOneExam()` that gets the scores of the i'th exam for all the students
  - Same as getting a column in a 2D array
- Write a method call `printEverything()` that prints all the grades for each student in the class