

Multi-Dimensional Arrays II

- Extend the program to print out the matrix.

```
class Trans {  
    public static void print (int[][] a, int len) {  
        for(int i = 0; i < len; i++) {  
            for(int j = 0; j < len; j++)  
                System.out.print(a[i][j] + "\t");  
            System.out.println();  
        }  
        System.out.println();  
    }  
    public static void main (String args[]) {  
        int a [][] = random(8);  
        print(a, 8);  
    }  
}
```

Slide 9-2



University of
Auckland

415.101

Java

Christian Collberg
September 28, 1997

Tutorial #9

Copyright © 1997 C. Collberg

Multi-Dimensional Arrays III

- Extend the program to also transpose the matrix, ie. rows and columns are interchanged.

```
class Trans {  
    public static int[][] transpose (int[][] b, int len) {  
        int[][] a = new int[len][len];  
        for(int i = 0; i < len; i++)  
            for(int j = 0; j < len; j++)  
                a[i][j] = b[j][i];  
        return a; }  
    public static void main (String args[]) {  
        int a [][] = random(8); print(a, 8);  
        a = transpose(a, 8); print(a, 8);  
    }  
}
```

Slide 9-3

Multi-Dimensional Arrays I

- Write a program which constructs an 8×8 integer matrix of random values $0 \dots 9$.

```
class Trans {  
    public static int[][] random (int len) {  
        int[][] a = new int[len][len];  
        for(int i = 0; i < len; i++)  
            for(int j = 0; j < len; j++)  
                a[i][j] = (int) (Math.random() * 10.0);  
        return a; }  
    public static void main (String args[]) {  
        int a [][] = random(8);  
    }  
}
```

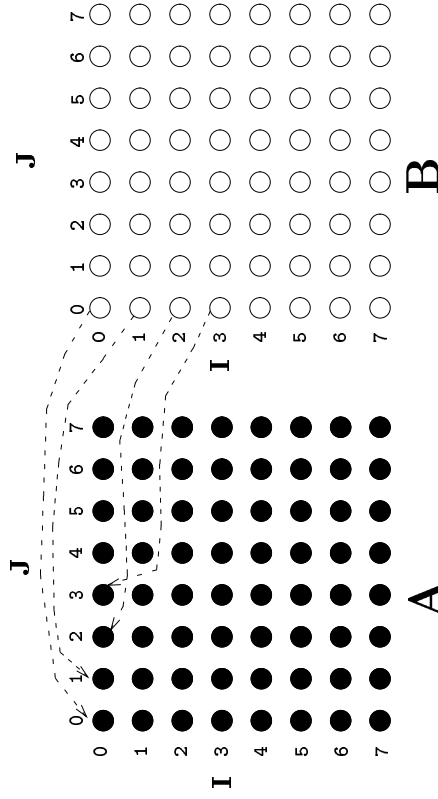
Slide 9-1

Class CharDisplay I

```
class CharDisplay {
    public final static int LEFT = 0;
    public final static int RIGHT = 1;
    public final static int CENTER = 2;
    public CharDisplay( int maxRow, int maxCol ) {...}
    // Write out the string 's'.
    public void write(
        int row, int col, int alignment, String s ) {...}
    // Write out the integer 'value'.
    public void write(
        int row, int col, int alignment, int value ) {...}
    // Flush the character display image to the screen
    public void flush() {...}
}
```

Slide 9-6

Multi-Dimensional Arrays IV



Slide 9-4

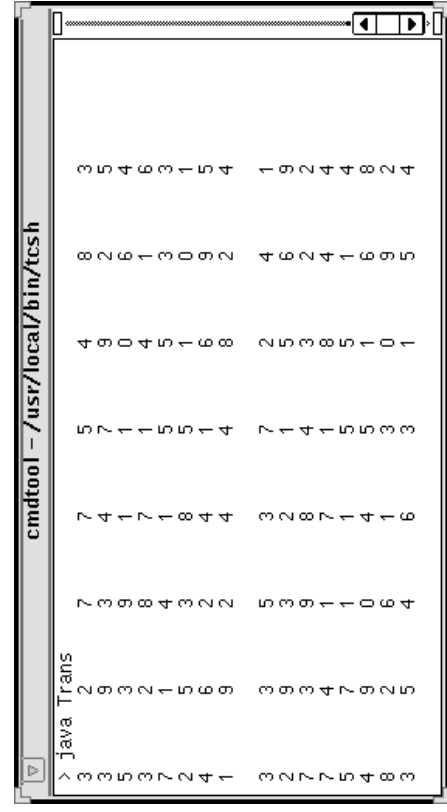
Class CharDisplay II

- Use class CharDisplay to graph the Math.sqrt function. Remember that in CharDisplay (0,0) is in the *top* left corner, but when graphing we want (0,0) in the *bottom* left corner.

```
class Sqrt {
    public static void main (String args[]) {
        CharDisplay d = new CharDisplay(10,60);
        for(int i = 0; i < 60; i++)
            d.write(9-(int)Math.round(Math.sqrt((double)i)), i, 0, "*");
        d.flush();
    }
}
```

Slide 9-7

Multi-Dimensional Arrays V



Class CharDisplay III

```

cmdtool - /usr/local/bin/tcsh
> java Sqrt
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

Slide 9-10

Fib

- What is the output of the program below?

```

class Fib {
    public static int F (int n) {
        int[] a = new int[n+1];
        a[0] = a[1] = 1;
        for(int i = 2; i <= n; i++)
            a[i] = a[i-1] + a[i - 2];
        return a[n];
    }
    public static void main (String args[]) {
        System.out.println(F(10));
    }
}

```

```

shelltool - /usr/local/
> javac Fib.java
> java Fib
89
>

```

Slide 9-11

Class CharDisplay III

```

shelltool - /usr/local/bin/tcsh
> java Sqrt
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55
*****
*****
*****
*****
*****
*****
*****
*****
*****
*****

```

Slide 9-8

Class CharDisplay III

- Extend the Sqrt class to also print out the X- and Y-axis.

```

class Sqrt1 {
    public static void main (String args[]) {
        CharDisplay d = new CharDisplay(11,64);
        for(int i = 0; i < 60; i++)
            d.write(9-(int)Math.round(Math.sqrt((double)i)), i+3,0,"");
        for(int i = 0; i < 60; i+=5)
            d.write(10, i+3, 0, i);
        for(int i = 0; i < 10; i++)
            d.write(9-i, 0, 0, i);
        d.flush();
    }
}

```

Slide 9-9

Errors

- Find all errors in the program below!

```
class Err {
    public static void F (int n) {
        int a = new int[n];
        a[0] = a[1] = 1;
        for(int i = 2; i < n; i++)
            a[i] += a[i-1]
        return a[n];
    }
    public static void main (String args[]) {
        System.out.println(F(1.0));
    }
}
```

Slide 9–12

```
class Err {
    public static void F (int n) {
        ^ Should return an int.

        int a = new int[n];
            ^ Missing brackets.

        a[0] = a[1] = 1;
        for(int i = 2; i < n; i++)
            a[i] += a[i-1]
                ^ Missing semicolon

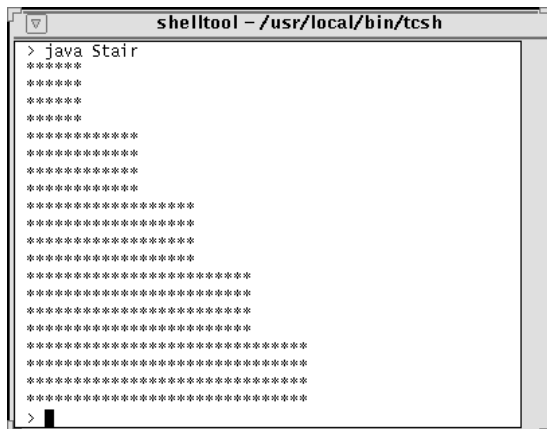
        return a[n];
            ^ Only a[0]...a[n-1] are valid array elements.
    }

    public static void main (String args[]) {
        System.out.println(F(1.0));
            ^ Can't pass a double to an int.
    }
}
```

Slide 9-13

Stairs I

- Write a program that prints out the flight of stairs below!



Slide 9-14

Stairs II

```

class Stair {
    static final int NumberOfSteps = 5;
    static final int StepHeight = 4;
    static final int StepLength = 6;

    public static void main (String args[]) {
        for(int i = 0; i<NumberOfSteps; i++)
            for(int h=0; h<StepHeight; h++) {
                for(int s=0; s<(StepLength*(i+1)); s++)
                    System.out.print("#");
                System.out.println();
            }
        }
    }
}

```

Slide 9–15

Fill I

- Write a program that fills a 5×10 array with the number $0 \dots 49$ and the prints out the array. Use div (/) and mod (%) to calculate the position of each entry!

```

> java Fill
0 1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48 49
>

```

Slide 9-16

Fill II

```

class Fill {
    static final int ROWS = 5;
    static final int COLS = 10;
    public static void main (String args[]) {
        int [][] d = new int [ROWS][COLS];
        for(int i = 0; i < (ROWS*COLS); i++)
            d[i / COLS][i % COLS] = i;
        for(int i = 0; i < ROWS; i++) {
            for(int j = 0; j < COLS; j++)
                System.out.print(d[i][j] + "\t");
            System.out.println();
        }
    }
}

```

Slide 9-17

Method Call I

- What's the output of the program below?

```

class Call {
    public static int F (int x) {
        int y = 1;
        for(int i = 0; i < x; i++)
            y = y * x;
        return y; }
    public static int G (String x) {
        int y = 1;
        for(int i = 0; i < x.length(); i++)
            y += H(i);
        return y; }
    public static int H (int x) {
        int y = 1;
        while (x > 0) {
            y = x * 2; x--;
        };
        return y; }
    public static void main (String args[]) {
        System.out.println(F(3) + G("Bye")); }
}

```

Slide 9-18

Method Call II

```

F(3)    = 1 * 3 * 3 * 3
        = 27
G("Bye") = 1 + H(0) + H(1) + H(2)
        = 1 + 1 + 2 + 2
        = 6
H(0)    = 1
H(1)    = 1 * 2
H(2)    = 1 * 2 * 1
F(3) + G("Bye") = 27 + 6 = 33

```

```

> java Call
33
>

```

Slide 9-19