

As you arrive:

1. Start up your computer and plug it in
2. **Log into Angel** and go to CSSE 120
3. Do the **Attendance Widget** – the PIN is on the board
4. Go to the course **Schedule Page**
5. Open the **Slides** for today if you wish
6. Check out today's first project:

Session28_2DArrays

Plus in-class time working on these concepts AND practicing previous concepts, continued as homework.

Final exam

- Date, time, and location
- C and Python
- Resources that you need

2D Arrays and Files

- Declaring, initializing, and using 2D arrays
- Opening files for reading, writing, appending
- Remember to close your files

Final Exam Facts

- **Date:** Thursday, November 18, 2010
- **Time:** 8:00 a.m. to noon
- **Venue:** **See schedule lookup page or course schedule**
- **Chapters:** Zelle chapters 1 to 12.1, Assigned C readings from Kochan or Web resources linked from course schedule and course resources page
- You may bring **two sheets** of paper this time.
- **Note that the C material will be emphasized, but comparing and contrasting C and Python will definitely be on the exam.**

Two-dimensional Arrays

- Like a list of lists in Python
- But size is fixed, like C arrays
- Visualize as a matrix:

NUM_ROWS = 3

NUM_COLS = 6

4	3	6	31	8	2
9	4	7	8	4	1
34	2	16	5	3	6

- Can make ragged arrays (different number of items in each row) but more difficult to do

2D Array Syntax

Checkout today's first project:
[Session28_2DArrays](#)

- Declaration reserves space, **but doesn't set values** to anything!

```
int nums[NUM_ROWS][NUM_COLS];
```

- **Looping through the array** (to display its values)

```
for (i = 0; i < NUM_ROWS; i++) {  
    for (j = 0; j < NUM_COLS; j++) {  
        printf("%2i ", nums[i][j]);  
    }  
    printf("\n");  
}
```

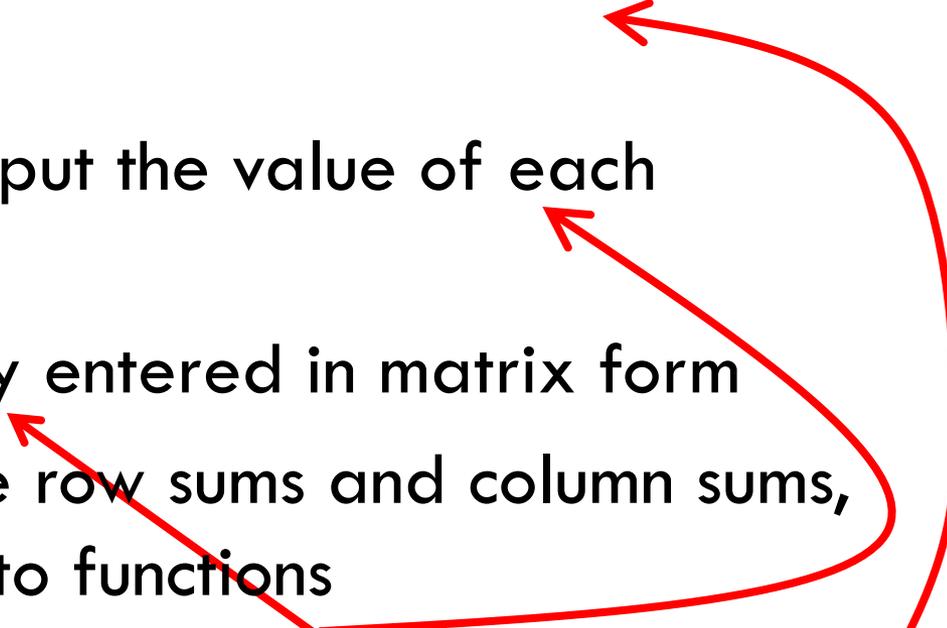
NUM_ROWS = 3
(outer loop, using *i*)

NUM_COLS = 6 (inner loop, using *j*)

4	3	6	31	8	2
9	4	7	8	4	1
34	2	16	5	3	6

Do TODO #1, 2, and 3 in today's project

Modify your code

- Ask the user for the number of rows and columns instead
 - Then prompt them to input the value of each element
 - Print out the values they entered in matrix form
 - Challenge: print out the row sums and column sums, passing the 2D arrays to functions
- 

Optional challenge problem:
Implement **example2** in today's
project, by doing TODO #5, 6 and 7.

Do TODO #4 in today's project.
It asks you to do the above.

File handling

- Need to include `<stdlib.h>` to access many file handling functions
- Open a file using `fopen (...)`
 - ▣ Takes the name of the file to open and the mode in which the file is to be opened
 - ▣ Modes:
 - "r" (read)
 - "w" (write)
 - "a" (append)
 - ▣ Returns a *file pointer* to access the file, of type: `FILE*`
- Close a file using `fclose (...)`
 - ▣ Takes the file pointer to close

Example on next slide

A simple example

```
FILE *inFile;  
inFile = fopen("my_file.txt", "r");  
if (inFile == NULL) {  
    exit(EXIT_FAILURE);  
}
```

How to read and write is explained on the next slides

```
// Read data from the file pointed to by inFile
```

```
fclose(inFile);
```

How do we read from a file?

- `getc(my_fileptr); /* read and return the next character from the file */`
- `fgets(buffer, n, my_fileptr);`
`/* read the next line of text from the file, up to n-1 characters, into buffer */`
- `fscanf(my_fileptr, "%i", &num);`
`/* read the next int value from the file into variable num */`

How do we write to a file?

- `putc(c, my_fileptr); /* Converts int c to a char and writes it to file */`
- `fputs(my_string, my_fileptr);`
`/* Copies my_string to file, except for the string terminating char */`
- `fprintf(my_fileptr, "%s\n", my_string) ;`
`/* Similar to printf() except the first parameter is a file pointer */`

File Handling

- Check out today's second project:
Session28_FilesDemo
from your SVN repository.
- See problem description in comments
- Work on solving problem for 10 minutes

HW28

- See instructions linked from Schedule page.
- You will read two 2D matrices from a file, perform math on them and then output the result back to a file.