CS221
C and Systems
Programming

College student's most precious resource…

THE EISENHOWER MATRIX

	URGENT	NOT URGENT
IMPORTANT	DO IT FIRST	SCHEDULE IT
NOT IMPORTANT	DELEGATE IT	DELETE IT

…is TIME
Train yourself to be productive: 50 minutes of focused activity, 10 minute break (not longer!). Keep up for 4-5 hours eventually. By senior year, 80 minutes of focused activity, 10 minute break. Jot down distractions for "next break"

lvalues and rvalues

What are lvalues and rvalues?

rvalues can be use on the <u>right</u> of the '=' (assignment operator)
- `float val; // a variable`
- `3.145; // a constant`
- `(4 + 5); // an expression`
- `getValue(); // a method return value`
- `&someValue; // address of a variable`

lvalues can be use on the <u>left</u> of the '=' (assignment operator)

```
long idNum; // a variable

*myPointer; // 'referred-to' of a pointer
```

…and why do we care?

Well, we can't use an rvalue on the left of an '='

We can use lvalues as rvalues

We can only take the address of an lvalue: `&myVariable`
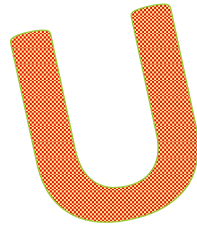
```
int x;
```

What is x?

'x' is an lvalue and an rvalue!
'x' could mean the (1) VALUE stored in x, or (2) it could be the STORAGE LOCATION we named 'x'.
You have to think about which meaning you are using: x = x + 1

More new C syntax

U

C union

```
union packet {
  struct {
    short IdValue
    short address
    short controlFlags;
    unsigned char payload[154];
  } header;
  int intData[40];
  unsigned char byteData[160];
};
```

7

All top-level members SHARE THE SAME MEMORY, SHARE THE SAME VALUES IN MEMORY.

Why would we ever do this?

Originally, to save memory back when memory was scarce

Gives us multiple different WAYS OF INTERPRETING the same values in memory

I have never written a union, but I had to use code that does

Do an example: put values in using 1 interpretation, read them out using a different one
LOOK AT uniondemo.c

Project:
Manala2

Review Project PDF
What are smart strategies to win at Mancala?

Mancala Solution Review

What are some key parts of the solution?

Build a board, print the board, free the board
- What features do we want in the board? Next square? Possible-capture square?
Find scoring bin?
Check if game over, calculate scores
Get players' inputs, validate for legality
Make a move, see if player gets to go again, handle capturing
TESTING: test input files, with "good" and "illegal" moves