## C: structs, malloc, free

CSE 413, Autumn 2007 10-24-2007

### Topics

- structs
- malloc, free and the heap

#### Structs

- A struct is a record. (similar to a Java object with no methods.)
  - » x.f is for field access.
  - » (\*x).f in C is like x.f in Java.
  - » x->f is an abbreviation for (\*x).f.
- There is a huge difference between passing a struct and passing a pointer to a struct.
- (see struct example code)

 $p \rightarrow x = new x;$ 

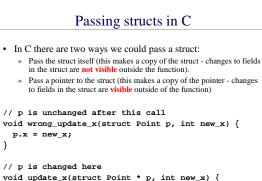
}

# Review: Passing Objects in Java Remember Java: objects are passed "by reference" (a copy of a *reference* to the object is passed)

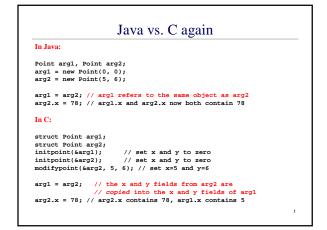
public static void java\_example(Point arg1, Point arg2) {
 // Modifies values referred to by arg1
 arg1.x = 100;
 arg1.y = 100;

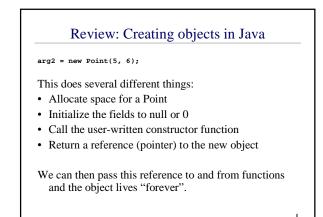
// Modifies the arguments
// (which are copies - does nothing)
Point temp = arg1;
arg1 = arg2;
arg2 = temp;
}
Notes:

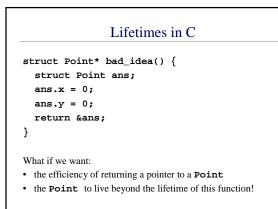
This does not make copies of the Point objects themselves, just a copy of the references (this is really like an address or pointer in C)



Returning structs in C
In C there is another way we could change a struct
inside of a function even if we pass in the struct itself:
// Modifies its own copy of p1, but then
// returns that modified copy.
struct Point change\_point(struct Point p1) {
 pl.x = 15;
 pl.y = 26;
 return p1; // p1 is copied back
}
This works, but copying entire structs can be less efficient than
 copying a single pointer/address.



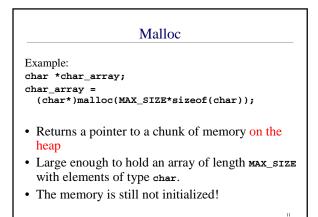






#### void \*malloc(size\_t size);

- **size\_t** is an unsigned long, indicates how many bytes of memory are requested.
- Returns a pointer to the newly-allocated memory.
- Returns NULL on failure.
- · Does not initialize the memory.
- You should cast the result to the pointer type you want.



Free	
<ul> <li>void *free(void* ptr);</li> <li>Returns the chunk of memory pointed to by ptr to the heap.</li> </ul>	
Example:	
<pre>int *buffer1; buffer1 = (int*) malloc(50*sizeof(int)); free(buffer1);</pre>	
You should free what you malloc. Why?	
Q: What is the value of buffer1 now?	