



Pointer Summary

The Finer Points

CS2263 – Systems Software Development




1

Just Pointers

1. A pointer is an address variable
2. The '&' operator returns the address of a variable
3. The '*' operator returns the value at the address held by the variable


```
int a; //declares an integer
int* pa; //declares an integer pointer
pa = &a; // gets the address of a and stores it in pa
        // also note that the datatypes on either side
        // of the equal sign must be the same
```

 - A. pa is an int*
 - B. a is an int
 - C. *pa is an int
 - dereference: get the value stored at the address in pa
4. Passing a pointer to a function means that the dereferenced value can be altered.
 - A. Address is passed-by-value; the value at that address can be altered



2

Pointers and Arrays

1. An array name is a pointer to the beginning of the array.
 - A. If declared on the stack (e.g. `int arr[10]`), then its address cannot be changed
 - no pointer to this stack memory should be returned from the function that declares it
 - Any function above this function in the stack (called by it) can use it though
 - B. If declared on the heap (e.g. `int* arr = (int*)malloc(sizeof(int)*10)`, then it can be changed.
 1. BUT changing it is a bad idea since it's the "anchor" pointer to the array
 - C. `arr == &arr[0] // int* == int*`
 - D. `*arr == arr[0] // int == int`
2. Passing an array to a function passes the address of the array.



3

Pointers and Strings

1. A string is an array (see all of the above) that has an extra character to hold a terminating character (NULL)
 - A. Allows us to not have to track the length of the string
 - B. Need to account for this when allocating memory on the heap


```
char* s = (char*)malloc(sizeof(char)*(n+1) );
```
2. Pointers allow us to substring


```
char* s = "I love C!";
printf("%s\n", s);
printf("%s\n", &s[0]);
printf("%s\n", &s[2]);
```

```
[FCSSSDs-iMac-3:Assignments/A3/src] wightman% a.out
I love C!
I love C!
love C!
[FCSSSDs-iMac-3:Assignments/A3/src] wightman% █
```



4