

# Arrays w/ Pointers

Last updated 5/18/21

# Arrays w/ Pointers

- Pointers and Arrays

- Reminder: the name of an array is actually a pointer to the 0<sup>th</sup> element of the array

`int myArray[ ];` // myArray holds the value 0x1000 (ptr)

`myArray + 2` evaluates to 0x1008 (ptr arithmetic)

`*(myArray + n)` is equivalent to `myArray[n]`

↑  
pointer arithmetic

# Arrays w/ Pointers

- Pointers and Arrays

```
int Student[5];  
int* myPtrA;  
int* myPtrB;  
int* myPtrC;
```

```
myPtrA = &Student[2];
```

```
myPtrB = &Student[1] + 1;
```

```
myPtrC = Student
```

```
*myPtrA
```

```
*myPtrB
```

```
*(myPtrC + 2)
```

```
*(Student + 2)
```

		Addr
Student	2	0x1000
	3	0x1004
	4	0x1008
	7	0x100C
	6	0x1010
	5	0x1014
	0	0x1018

# Arrays w/ Pointers

- Pointers and Arrays

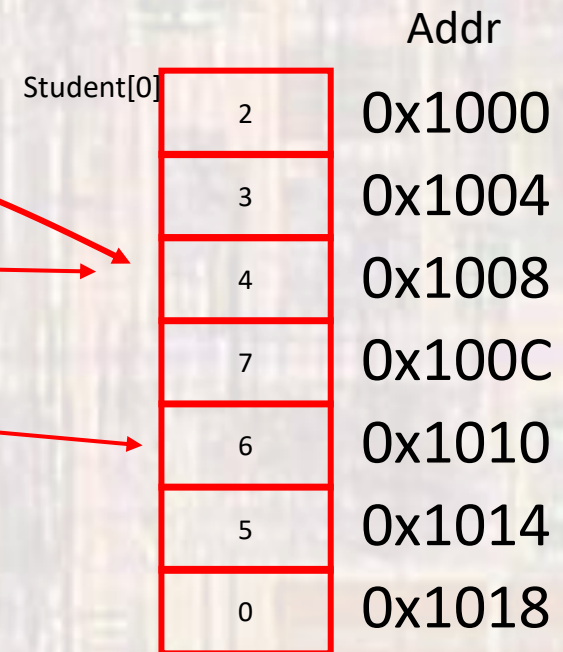
```
int Student[5];  
int* myPtrA;  
int* myPtrB;  
int* myPtrC;
```

```
1008 1008  
myPtrA = &Student[2];
```

```
1008 1004 + 4  
myPtrB = &Student[1] + 1;
```

```
1010 1000 + 10 (hex)  
myPtrC = Student + 4;
```

```
*myPtrA      4  
*myPtrB      4  
*(myPtrC - 2) 4  
*(Student + 2) 4
```

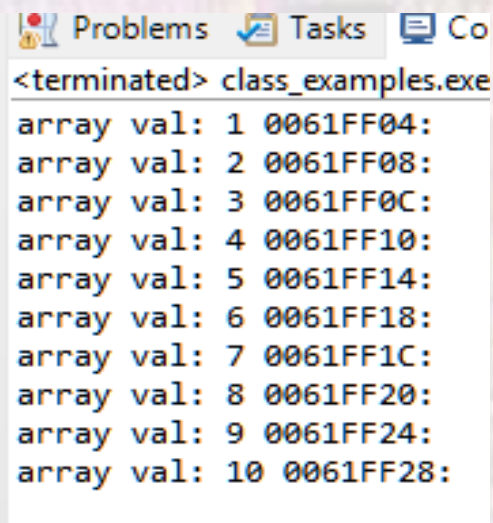


# Arrays w/ Pointers

- Pointers and Arrays

```
// Local variables
int myArray[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

// print array
int i;
for(i = 0; i < 10; i = i + 1){
    printf("array val: %i %p: \n", *(myArray + i), myArray + i);
}
```



```
Problems Tasks Co
<terminated> class_examples.exe
array val: 1 0061FF04:
array val: 2 0061FF08:
array val: 3 0061FF0C:
array val: 4 0061FF10:
array val: 5 0061FF14:
array val: 6 0061FF18:
array val: 7 0061FF1C:
array val: 8 0061FF20:
array val: 9 0061FF24:
array val: 10 0061FF28:
```

# Arrays w/ Pointers

- Pointers and Arrays

```
// Local variables
double myArray2[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

// print array
int i;
for(i = 0; i < 10; i = i + 1){
    printf("array val: %f %p: \n", *(myArray2 + i), myArray2 + i);
}
```

```
array val: 1.000000 0061FEA0:
array val: 2.000000 0061FEA8:
array val: 3.000000 0061FEB0:
array val: 4.000000 0061FEB8:
array val: 5.000000 0061FEC0:
array val: 6.000000 0061FEC8:
array val: 7.000000 0061FED0:
array val: 8.000000 0061FED8:
array val: 9.000000 0061FEE0:
array val: 10.000000 0061FEE8:
```

# Arrays w/ Pointers

- Pointers and Arrays

```
// Local variables
char myArray3[10] = {49, 50, 51, 52, 53, 54, 55, 56, 57, 58 };

// print array
int i;
for(i = 0; i < 10; i = i + 1){
    printf("array val: %c %p: \n", *(myArray3 + i), myArray3 + i);
}
```

```
array val: 1 0061FE96:
array val: 2 0061FE97:
array val: 3 0061FE98:
array val: 4 0061FE99:
array val: 5 0061FE9A:
array val: 6 0061FE9B:
array val: 7 0061FE9C:
array val: 8 0061FE9D:
array val: 9 0061FE9E:
array val: : 0061FE9F:
```

# Arrays w/ Pointers

- Pointers and Arrays – example 1
  - The pointer terminology can replace our array terminology

```
////////////////////////////////////
// arraypointers_class_ex_1 project
// created 5/12/21 by tj
// rev 0
//
////////////////////////////////////
// Using pointers to access arrays example file for class
// Prints an array backwards
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

#define N 5

int main(void){
    setbuf(stdout, NULL); // fix for terminal issue

    // splash
    printf("\n\narraypointers_class_ex_1 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
           MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // local variables
    int myArray[N];
    int* ary_ptr;

    // read the array
    printf("Please enter %i integer array values: ", N);
    for(ary_ptr = myArray; ary_ptr < myArray + N; ary_ptr++){
        scanf("%i", ary_ptr);
    }

    // print array backwards
    printf("Your array backwards is: ");
    for(ary_ptr = myArray + (N-1); ary_ptr >= myArray; ary_ptr--){
        printf("%i ", *ary_ptr);
    }

    return 0;
} // end main
```

```
arraypointers_class_ex_1 - example for EE2905
Using Mbed OS version 6.10.0

Please enter 5 integer array values: 22 33 44 55 66
Your array backwards is: 66 55 44 33 22 █
```



# Arrays w/ Pointers

- Pointers and Arrays – example 2
  - The pointer terminology can replace our array terminology

```
////////////////////////////////////
//
// arraypointers_class_ex_2 project
//
// created 5/12/21 by tj
// rev 0
//
////////////////////////////////////
// Using pointers to access arrays example file for class
//
// Finds largest value in the array
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

#define N 5

// Function Prototypes (Declarations)
int largest(int* ary, int n);

int main(void){
    setbuf(stdout, NULL); // fix for terminal issue

    // splash
    printf("\n\narraypointers_class_ex_2 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n",
           MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // local variables
    int myArray[N];
    int* ary_ptr;
    int tmp;

    // read the array
    printf("Please enter %i integer array values: ", N);
    for(ary_ptr = myArray; ary_ptr < myArray + N; ary_ptr++){
        scanf("%i", ary_ptr);
    }
}
```

```
// find largest
tmp = largest(myArray, N);

// print largest
printf("The largest value in your array is: ");
printf("%i ", tmp);

return 0;
} // end main
```

```
// Function Definitions
int largest(int* ary, int n){
    // local variables
    int i;
    int large;

    large = *ary; // set to first value

    // find largest
    for(i = 1; i < n; i++){
        if(*(ary + i) > large)
            large = *(ary + i);
    }

    return large;
} // end largest
```

```
arraypointers_class_ex_2 - example for EE2905
Using Mbed OS version 6.10.0
```

```
Please enter 5 integer array values: 5 7 9 4 2
The largest value in your array is: 9
```