

# Scope

Last updated 9/7/21

# Scope

- Scope
  - Region of a program in which a defined object is visible
  - Defined Objects
    - Variables
    - Functions
  - Two types of regions
    - Blocks
    - Not in a block

# Scope

- Program Prototype

- Blocks

- Statements enclosed in { ... }

- Contents of Main
    - Contents of Functions

- Not in a Block

- Global Area

```
// comments
```

```
#include <stdio.h>  
int foo;
```

Global Area

```
int fun1(int x, int y); // function prototype
```

```
int main(void){
```

```
int x;
```

```
int y;
```

```
float a;
```

```
if(...){
```

```
float x;
```

```
float a;
```

```
float b;
```

```
x = a * 3
```

```
}
```

```
else
```

```
a = x * y;
```

```
...
```

```
} // end of main
```

Main's Area

Nested Block Area

```
int fun1 (int i, int j){
```

```
int x;
```

```
int y;
```

```
...
```

```
} // end of fun1
```

Function fun1 Area

# Scope

- Scope Extents
  - An objects scope extends from it's declaration to the end of it's block
- Global Scope
  - Any object defined in the global area of a program
  - Visible anywhere in the current program
- Local Scope
  - Any object defined in a block area
  - Includes Main and Functions
  - Visible anywhere in the current block (after it's declaration)

# Scope

- Examples
  - Local definitions within a block supersede higher level definitions

```
// example
```

```
#include <stdio.h>
```

```
int x;
```

```
int y;
```

```
int main(void){
```

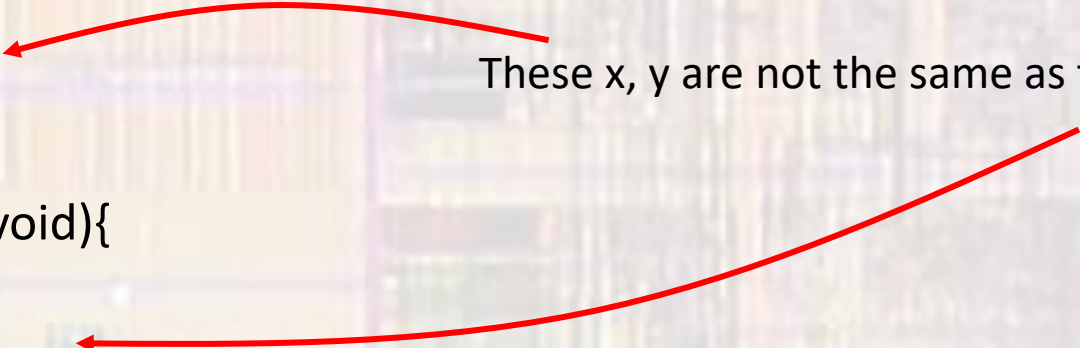
```
    int x;
```

```
    float y;
```

```
    ...
```

```
}
```

These x, y are not the same as these





# Scope

- Examples

```
// comments
#include <stdio.h>
Int foo;

int fun1(int x, int y);    // function prototype

int main(void){
  int x;
  int y;
  float a;
  if(...){
    float x;
    x = a * 3;
    float a;
    float b;
  }
  else
    b = x * y;
  ...
} // end of main

int fun1 (int i, int j){
  int x;
  int y;
  ...
} // end of fun1
```

Annotations:

- foo is visible here (points to the `Int foo;` line)
- this a is visible here but this is a new a (points to the `float a;` line in `main` and the `float a;` line in the `if` block)
- new i,j new x,y only visible in fun1 (points to the `int i, int j` and `int x, int y` lines in `fun1`)

# Scope

- Examples

```
////////////////////////////////////
//
// scope_class_ex_1 project
//
// created 5/12/21 by tj
// rev 0
//
////////////////////////////////////
//
// scope example file for class
//
// Taken from Dr. Widder
//
////////////////////////////////////
#include "mbed.h"
#include <stdio.h>

// Function Prototypes (Declarations)
int vegas(int i, int j);

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

    // splash
    printf("\n\nscope_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 i;
    int j;
    int k;
    i = 2;
    j = 4;
    k = 0;

    printf("i = %i, j = %i, k = %i \n", i, j, k);

    k = vegas(i, j);

    printf("i = %i, j = %i, k = %i \n", i, j, k);

    return 0;
} // end main
```

```
// Function Definitions
int vegas(int i, int j){
    // special vegas fn

    // local variables
    int foo;
    foo = 0;

    // update values
    i++;
    j++;
    foo = i * j;

    return foo;
} // end vegas
```

# Scope

- Examples

```
////////////////////////////////////
//
// scope_class_ex_1 project
//
// created 5/12/21 by tj
// rev 0
//
////////////////////////////////////
//
// scope example file for class
//
// Taken from Dr. Widder
//
////////////////////////////////////
#include "mbed.h"
#include <stdio.h>

// Function Prototypes (Declarations)
int vegas(int i, int j);

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

    // splash
    printf("\n\nscope_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 i;
    int j;
    int k;
    i = 2;
    j = 4;
    k = 0;

    printf("i = %i, j = %i, k = %i \n", i, j, k);

    k = vegas(i, j);

    printf("i = %i, j = %i, k = %i \n", i, j, k);

    return 0;
} // end main
```

```
// Function Definitions
int vegas(int i, int j){
    // special vegas fn

    // local variables
    int foo;
    foo = 0;

    // update values
    i++;
    j++;
    foo = i * j;

    return foo;
} // end vegas
```

```
scope_class_ex_1 - example for EE2905
Using Mbed OS version 6.10.0
```

```
i = 2, j = 4, k = 0
i = 2, j = 4, k = 15
```



# Scope

- Static Variables
  - Hold their value even after their scope has ended

```
////////////////////////////////////
//
// scope_class_ex_2 project
//
// created 5/12/21 by tj
// rev 0
//
////////////////////////////////////
//
// scope example file for class
//
// Shows static variables hold their value
//
////////////////////////////////////

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

// Function Prototypes (Declarations)
int fun1(void);
int fun2(void);

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

    // splash
    printf("\n\nscope_class_ex_2 - 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

    // call each function 3 times
    printf("%d ", fun1());
    printf("%d ", fun1());
    printf("%d ", fun1());
    printf("\n");
    printf("%d ", fun2());
    printf("%d ", fun2());
    printf("%d ", fun2());

    return 0;
} // end main
```

```
// Function Definitions
int fun1(void){
    // function 1

    // local variables
    int count;
    count = 0;

    count++;

    return count;
} // end fun1

int fun2(void){
    // function 2

    // static variable
    static int count = 0; // special case for static vars

    count++;

    return count;
} // end fun2
```

**scope\_class\_ex\_2 - example for EE2905  
Using Mbed OS version 6.10.0**

```
1 1 1  
1 2 3
```