Static Variables

Last updated 5/15/24

These slides describe static variables in C

Static Variables

- Static variables hold their value inside a scope even after their scope has ended
- Typically used in functions
  - Keep track of how many times we entered the function
  - Keep track of how long since we last entered the function

Static Variables

- Format:

```
static type variable_name = value
```

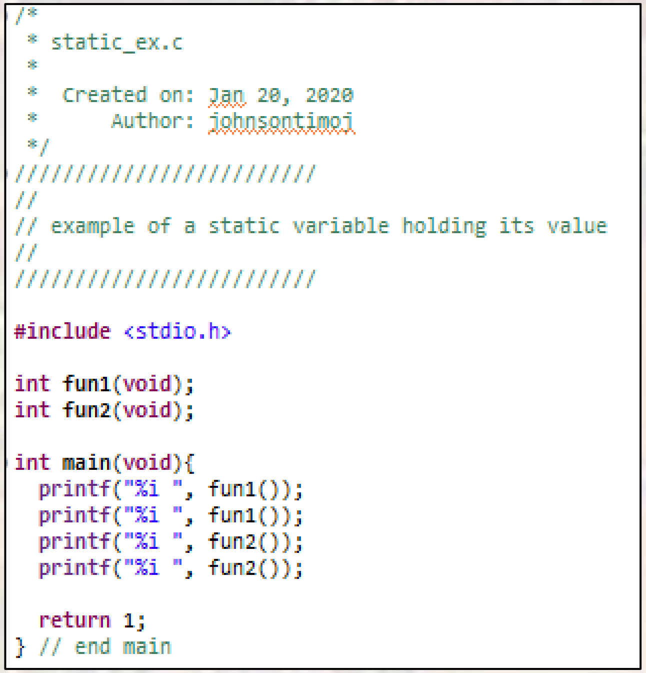
- This is an exception to our no declaration and initialization on the same line rule

- The initialization only occurs the first time the declaration is executed.

- Stored separately in Data Memory
  - Not in the Stack

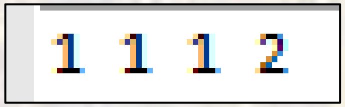
Static Variables

- Example

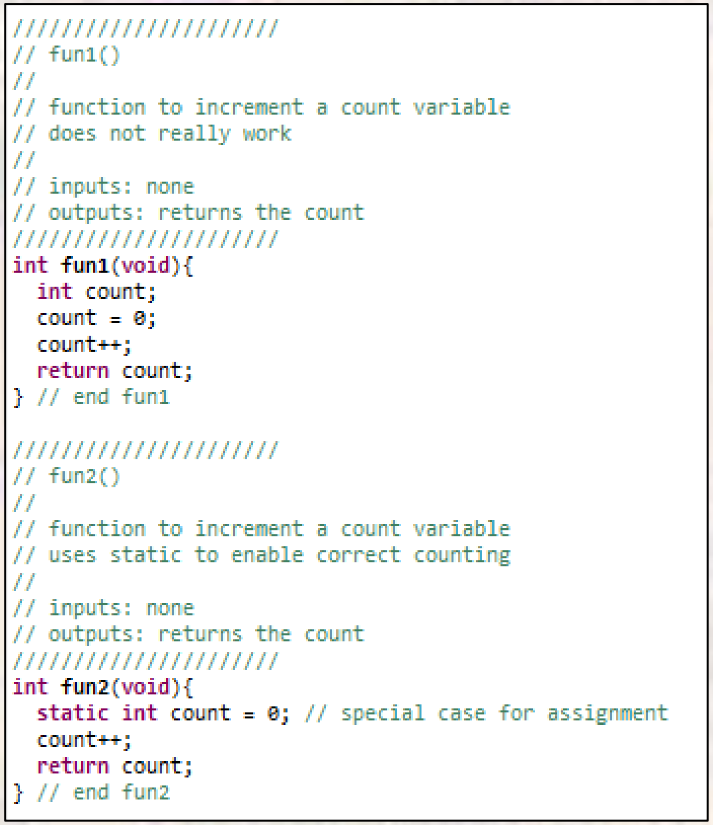


```c
/*
 * static_ex.c
 *
 *  Created on: Jan 20, 2020
 *      Author: johnsontimoj
 */
/////////////////////////////
//
// example of a static variable holding its value
//
/////////////////////////////

#include <stdio.h>

int fun1(void);
int fun2(void);

int main(void){
  printf("%i ", fun1());
  printf("%i ", fun1());
  printf("%i ", fun2());
  printf("%i ", fun2());

  return 1;
} // end main
```

result



```
1 1 1 2
```



```c
//////////////////////
// fun1()
//
// function to increment a count variable
// does not really work
//
// inputs: none
// outputs: returns the count
//////////////////////
int fun1(void){
  int count;
  count = 0;
  count++;
  return count;
} // end fun1

//////////////////////
// fun2()
//
// function to increment a count variable
// uses static to enable correct counting
//
// inputs: none
// outputs: returns the count
//////////////////////
int fun2(void){
  static int count = 0; // special case for assignment
  count++;
  return count;
} // end fun2
```

count recreated every time the function is called

always returns the value 1

count created once and saved for the next time the function is called

count increments each time the function is called