Week 3-4

- Control flow (review)
  - Conditional statements
    - If, else, else if, switch-case, break
  - Loop constructs
    - for, while, do-while, break, continue, label--go;
- Function definition
  - Return type, parameter definition, parameter passing
  - Function calls and standard library
- Program Structures
  - Preprocessing
  - Header files, Application Programming Interfaces (API)
  - Scope rules
  - C library

Conditional statements

- If, else, else if,
- switch-case, break
  - http://www.cse.ohio-state.edu/~yuw/459-21/switch.c

Control flow

- Loop constructs
  - for
  - while
  - do-while
- Other flow control methods
  - break
  - continue
  - label—go
- Sample programs
  - calculate the sum of 1-10 using different loops

Function

- Function prototype
  - return type
  - argument definition
- Function calls
  - basic syntax
  - parameter passing
  - Example: find the next prime number

Function Organization

- A set of functions
  - Defined in the same files as main()
    - Add two more interfaces:
      - my_isuppercase() and my_islowercase()
    - Sample program
  - Use separated file
    - Header files
      - Application Programming Interfaces (API)
    - How to modify the above sample
      - counting.c
      - new_ctype.h
      - new_ctype.c
  - Separated compilation

Recursion

- A function can call itself
  - Directly
  - Indirectly
  - examples:
    - Calculate the power of x to y
    - Find a matching number in an increasing array of integers

Scope Rules

- Scope:
  - Where the variable or function definitions can be accessed
- Local and global variables
  - Local: a variable defined inside a function
  - Global: a variable defined outside of functions
- external variable
  - Note: global variable and external variables
  - Typically used when a global variable is used in another file

Scope Rules -- continued

- static variable
  - Value are retained on exit from function
  - Limited scope
    - Only accessible within the same file
    - Or other source files that 'include's this file
- Automatic variables:
  - Typically local variables
  - variables that are automatically created/destroyed
- Register variable:
  - variables are saved in register for fast access

Function scope

- All functions are defined externally
- Static function, limited scope
- Inline functions

Preprocessing

- Preprocessing directives
  - file inclusion
    - #include
  - code replacement
    - #define
  - Others
    - #error, #warning
    - #pragma, machine or platform dependent
  - Conditional compilation
    - #ifdef -- #endif; #if EXP -- #endif
  - Conditional inclusion
    - #ifdef #include #endif

C library

- Library:
  - A collection of object files that contains the function definitions
- Static and shared libraries
  - Static library
    - Copies of function definitions are compiled into the final executable
  - Shared library
    - Functions are not copied into the final executable
    - Information are recorded in the image so that the correct library will loaded on demand when the program starts up.

C library -- continued

- Creating libraries on *nix systems
  - Static library, with suffix .a
    - gcc -c file1.o
    - ar cr libc459.a file1.o
  - Shared library, with suffix .so
    - gcc -c -fPIC file1.c
    - gcc -shared -fPIC -o libc459.so file1.o