C Learning Notes

Keep a place to summarize concepts from each lecture. This can be created in onenote, onedrive, google drive, website or other tool so it can be handed in for learning activities.

Here is an example of one format but you can do what works best for you.

[General Resources](#_wwbu7uriit6l)

[C Documentation](#_f6g2csgye3js)

[Getting Your Questions Answered](#_o8qxhnmspugm)

[Secure C Programming](#_t3k1c5divuyh)

[Avoid single-argument printfs](#_nd2h45wdg8m5)

[Setting up C Environment and Debugging](#_2qx79kcmn9e8)

[Other Developer Tools](#_3p2lbg6weaki)

[Chapter 2 Introduction to C Programming](#_3c1l82tf8lee)

[Standard Input/Output, printf/scanf format conversion specifiers](#_ud1ufgmaoih)

[Operators arithmetic, &, assignment = and relational ==, modular](#_jfpl62qtbiev)

[Errors Syntax and Runtime](#_bnivutg83cy8)

[Chapter 1 Introduction to Computers](#_nylfffei17ag)

[Typical phases of C Development Environment](#_pvc9jb2liozh)

[C programming Language](#_389ltnq3tfpq)

# General Resources

## C Documentation

* [C Standard Library header files](https://en.cppreference.com/w/c/header) - contains standard library headers and what functions they contain, the headers need to be added in preprocessor phase to use functions from the library in your code
* [C language](https://en.cppreference.com/w/c/language) This is a reference of the core C language constructs.
* [The GNU C Library Reference Manual](https://www.gnu.org/software/libc/manual/pdf/libc.pdf)
* [Microsoft C Language Reference](https://docs.microsoft.com/en-us/cpp/c-language/c-language-reference)

## Getting Your Questions Answered

Online forums enable you to interact with other C programmers worldwide and get your questions answered. Popular C and general programming online forums include:

* [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com)
* [r/C\_Programming: C](https://www.reddit.com/r/C_Programming/)
* [C Google (comp.lang.c)](https://groups.google.com/g/comp.lang.C)
* [C Board](https://cboard.cprogramming.com/c-programming/)
* [C and C++ Forum | Dream.In.Code](https://www.dreamincode.net/forums/forum/15-c-and-c/)
* [GeeksforGeeks Stuck in Programming: Get The Solution From These 10 Best Websites](https://www.geeksforgeeks.org/stuck-in-programming-get-the-solution-from-these-10-best-websites/)

## Secure C Programming

[SEI CERT C Coding Standard](https://wiki.sei.cmu.edu/confluence/display/c)

### Avoid single-argument printfs

For more information, see [CERT rule FIO30-C](https://wiki.sei.cmu.edu/confluence/display/c/FIO30-C.+Exclude+user+input+from+format+strings). Chapter 6’s Secure C Programming section explains the notion of user input as referred to by this CERT guideline.

[Format String Software Attack | OWASP Foundation](https://owasp.org/www-community/attacks/Format_string_attack).

p. 57-58 If you need to display a string that terminates with a newline, use the puts function, which displays its string argument followed by a newline character

Instead of **printf("Enter first integer\n");**

Write  **puts("Enter first integer\n");**

We did not include \n in the preceding string because puts adds it automatically.

If you need to display a string without a terminating newline character, use printf with two arguments.

Instead of **printf("Welcome ");**

Write **printf("%s", "Welcome ");**

## Setting up C Environment and Debugging

[Set up C Environment and Developer Resources](https://docs.google.com/document/d/1VTKdVXAq4DmK5aNcrvQB22RqOEHFSVhks9u_qwi2U84/edit#) - add specifics below for environment you are using

[Overview of Visual Studio](https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019)

[Xcode Help](https://help.apple.com/xcode/mac/current/) and [Xcode Documentation](https://developer.apple.com/documentation/xcode/)

How to create C project:

How to debug:

Setting breakpoints

Stepping through each line of code

Watching/Inspecting variables

## Other Developer Tools

[GitHub:](https://github.com/) Cloud Repository for Versioning Code

[ClangTidy](https://clang.llvm.org/extra/clang-tidy/)

[Docker: Empowering App Development for Developers](https://www.docker.com/)

Linux Commands

[Basic Linux Commands for Beginners | Linux](https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners)

To change folders use a wildcard character called \* as follows $ cd A\_VERY\_VERY\* Or type cd A\_VERY and hit tab key to complete a directory name $ cd A\_VERY\_VERY (hit tab key)

# Chapter 2 Introduction to C Programming

Code example from book: fig02\_02

## Standard Input/Output, printf/scanf format conversion specifiers

[File input/output](https://en.cppreference.com/w/c/io)

Escape Sequences p. 390

[Printf](https://en.cppreference.com/w/c/io/fprintf) p. 43

Conversion Specifiers printf p. 379 and scanf p.391-392

[Format specifiers in C](https://www.geeksforgeeks.org/format-specifiers-in-c/): place holder and conversion character

* Use % for place holder
  + Used here % is not a comment
  + % is holding a place to put the value of the variable
* conversion character for type
  + Used to specify the type of value to be printed
  + d is specifying a string will be printed in this location

 %d integer (decimal integer)

%f   float (real number)

%c character (one character)

%d string of characters

[Scanf](https://en.cppreference.com/w/c/io/fscanf)

This scanf has two arguments, "%d" and &integer1.

* %d conversion specifier indicates that the data should be an integer (the letter d stands for “decimal integer”).
* ampersand (&)—called the address operator

Code example from book fig02\_05

| // Fig. 2.5: fig02\_05.c  // Addition program  //preprocessor directive to read header files from standard library  #include <stdio.h> // adding standard input/output library  // function main begins program execution  int main( void )  {    int integer1; // first number to be entered by user  int integer2; // second number to be entered by user  //prompt for integers  //Do not use singlge argument printf  // printf("Enter first integer\n");  //use puts instead if want a newline  puts("Enter first integer\n");  // when reading integer include a conversion specifier  //%d indicate data should be integer (decimal integer)  //& in front of variable gives the location in memory to store the integer  scanf( "%d", &integer1 ); // read an integer  puts( "Enter second integer\n" ); // prompt  scanf( "%d", &integer2 ); // read an integer  int sum; // variable in which sum will be stored  sum = integer1 + integer2; // assign total to sum  if (integer1 == integer2)  {  printf("%d is equal to %d\n", integer1, integer2);  }  // can use printf because there is more than one argument  printf( "Sum is %d\n", sum ); // print sum  } // end function main |
| --- |

## Operators arithmetic, &, assignment = and relational ==, modular

[CPP: C Operator Precedence](https://en.cppreference.com/w/c/language/operator_precedence)

[Practical uses for the modulo operator — Federico Hatoum](https://hatoum.com/blog/2012/12/practical-uses-for-modulo-operator.html)

[Fun With Modular Arithmetic – BetterExplained](https://betterexplained.com/articles/fun-with-modular-arithmetic/)

## Errors Syntax and Runtime

A syntax error occurs when a compiler or interpreter cannot understand the source code statement in order to generate machine code. The syntax error is an incorrect construction of the source code

A runtime error occurs when the computer is running. For example, array out of bound, dividing by zero, accessing memory that is not available are some examples of runtime errors

# Chapter 1 Introduction to Computers

## Typical phases of C Development Environment

Phases p. 15

* Editor to create C code
* Preprocessor reads your header .h file(s)​ such as stdio.h
  + #include <stdio.h> is an example of a [preprocessor](http://preprocessor) directive in the source code
* Compiler processes your .c file​ and translates it into machine-language (object code). If no compile (syntax) errors then creates .o or .obj file​.
* Linker links the object code with the code for the missing referenced functions to produce an executable image. If the program compiles and links correctly, a file called a.out is produced.
  + For example
* Loader takes the executable image from disk and transfers it to memory and loads additional components for shared libraries.
* CPU executes the program one instruction at a time