COP 3223H: Introduction to C Programming

Fall 2023

University of Central Florida

Dr. Kevin Moran

Week 1 - Class2: C-Language Elements

UCF ACM Presentation

Administrivia

- Entrance Survey has been posted to Webcourses!
    - Due Friday at 11:59pm (should only take a few mins)
- Please sign up for Ed Discussions!

Today's Agenda

- Discuss C Language Elements and Syntax
- Go a bit deeper on our example program
- Discuss the bash command line

C Language Elements & Syntax

C Syntax

- Every Language has rules that you need to follow in order to express ideas, Programming Languages (PLs) are no different!
- Just as English has grammar, so does C, and we generally call this grammar its ***syntax.***
- For example, in C you will see that every statement ends in a semicolon.

Our First C Program

```c
// Simple C program to display "Hello World"

// Header file for input output functions
#include <stdio.h>

// main function –
// where the execution of program begins
int main()
{
    // prints hello world
    printf("Hello World \n");

    return 0;
}
```

Anatomy of a C Program

- Every C Program basically consists of the following parts:
  - Preprocessor Commands `#include <stdio.h>`
  - Functions `int main()`
  - Variables *We will cover next class!*
  - Statements & Expressions `printf("Hello World \n");`
  - Comments

```
// main function –
// where the execution of program begins
```

Tokens in C

- Tokens are one of the following:
  - keyword
  - identifier
  - constant
  - string literal
  - symbol

```
printf("Hello World \n");
```

Individual Tokens

```
printf
(
"Hello World \n”
)
;
```

Semicolons

- In a C program, the semicolon is a statement terminator
- Each individual statement must be ended with a semicolon, as it indicates the end of a logical entity.
- However, whitespace does not matter (I will demonstrate).

Comments

- Comments allow programmers to make notes about their code, and this is generally considered to be good practice.
- Code is often reused, updated, refactored, etc. Therefore, it is important for the author of a certain piece of code to make sure the intent is clear!
- In other words, it helps you to document the reason code was written or document a solution to the problem that the code solves.
- It also allows future coders who work on a past project to see the program intent.
- Syntax:

```
// This comment has one line

/* This comment has
many lines!!!*/
```

- Compilers completely ignore comments

Identifiers

- A C identifier is a name used to identify a variable, function, or any other user-defined item.
- An identifier starts with a letter A to Z, a to z, or an underscore '_' followed by zero or more letters, underscores, and digits (0 to 9).
- C does not allow punctuation characters such as @, $, and % within identifiers.
- C is a case-sensitive programming language.
  - Thus, `Manpower` and `manpower` are two different identifiers in C.
- It’s best to be consistent in your identifier scheme.
  - in this class, to keep things simple, we will use CamelCase for structs, and snake_case for everything else :-)

Keywords

- Keywords are reserved words that serve special functions that you cannot use for identifier names.

auto	else	long	switch
break	enum	register	typedef
case	extern	return	union
char	float	short	unsigned
const	for	signed	void
continue	goto	sizeof	volatile
default	if	static	while
do	int	struct	_Packed
double			

Anatomy of Hello World

```c
// Simple C program to display "Hello World"

// Header file for input output functions
#include <stdio.h>

// main function -
// where the execution of program begins
int main()
{
    // prints hello world
    printf("Hello World \n");

    return 0;
}
```

Preprocessor Directive

- Provides information to the preprocessor
- A preprocessor modifies a c program prior to its compilation
- stdio.h is the standard input/output header file
    - It contains pre-defined functions that we can use!

Anatomy of Hello World

```
// Simple C program to display "Hello World"

// Header file for input output functions
#include <stdio.h>

// main function -
// where the execution of program begins
int main()
{
    // prints hello world
    printf("Hello World \n");

    return 0;
}
```

Main Function

- C programs always execute instructions starting at the main function from top to bottom.
- All c programs are required to have a main function - otherwise ***syntax error.***
- The main function end with `return 0;`
  - This terminates the function (and program) by sending the value 0 back to the operating system of the computer.
  - Other values are used to indicate errors and should not be used!

Anatomy of Hello World

```
// Simple C program to display "Hello World"

// Header file for input output functions
#include <stdio.h>

// main function -
// where the execution of program begins
int main()
{
    // prints hello world
    printf("Hello World \n");

    return 0;
}
```

`printf()` Function

- This is a pre-defined function from the `stdio.h` library
- The function displays information to the user (and can also be useful for debugging)
- It displays text on lines
  - You have to specify the newline character `\n` to create a new line.

C Program Demo!

Command Line Basics

Command Line Overview

- In this class, when I refer to the command line, I am typically referring to the `bash shell`.
- Quite simply, bash is an interpreter that decodes commands that help us to do things on a computer via a text interface.
- Learning some simple command line basics will be invaluable in this course (and in the future).

Bash Commands

- `ls` - list directory contents
- `echo` - print text to the window
- `touch` - creates a file
- `mkdir` - creates a directory
- `grep` - search
- `cd` - change directory
- `pwd` - present working directory
- `mv` - move or rename directory/file
- `rm` - delete a file or directory
- `cat` - view the contents of a text file

Accessing the Terminal

- Mac, it is built in to the “Terminal.app” program!
- On Windows it is slightly more complicated.
  - The default command line on Windows is the “Windows Command Prompt”.
  - You will need to install “Windows Subsystem for Linux” to run bash.

Command Line Demo

Acknowledgements

Slides adapted from Dr. Andrew Steinberg’s COP 3223H course