Overview of C Programming

Alex Koschmann – Design Team 9 Team Manager

Texas Instruments Robotic Transportation Vehicle

Abstract

This application note has been written to give a review of the C programming language. Michigan State students have taken a one credit course covering this topic, and it is a sophomore level class, so most students have not had C programming experience in two years. The project assigned to the team has a majority of work to be done in C code, so an application note on the topic will help to remember everything that was learned previously. Major topics will include variables, loops, and input/output ability.
Background

The C programming language was developed in the early 1970s at AT&T Bell Labs. The language was made to replace the B programming language, which itself had several predecessors going all the way back to assembly language. C was developed to work with the Unix operating system. The language was mostly complete by 1973. In 1978, a book called “The C Programming Language” was published, which became the standard called K&R C. The second edition of the book covered ANSI C, the current standard.

The language is widely used for its broad capabilities and also its speed. It can be used in many different applications from writing simple programs to controlling powerful machines. It is among the fastest languages to use, as it works so closely with memory allocation. It has withstood the test of time, remaining one of the most popular languages to learn while remaining one of the oldest languages still used.

Variables

C variables must be explicitly declared to be used, unlike many simpler languages such as Python. Several data types are defined below:

char – Sets the variable as single character. It is an integer type, and can be signed or unsigned.

short – Signed integer type variable, sets a number at least 16 bits long.

int – A basic signed integer type. Variable is at least 16 bits in size.

long – Variable that is a signed integer. It is at least 64 bits long, first used in C99 version of standard.

float – Floating point type, single precision. Displays exact decimal values.

double – Double precision floating point type variable.

long double – extended precision floating point type. It is an 80 bit format usually.

_Bool – included in the stdbool.h library, defines true as 1 and false as 0.

Strings are a complicated task in C. Whereas other languages like python have a specified data type, C does not. It uses the char type, and uses an array of them to create the string. For example, the word “string” could be defined as char string[7], because there are 6 letters in the word, and a last character must be added, “\0”, to signify the end of the string.
Loops

There are three types of loops in C. An example of each is shown below.

```c
for (initialization variable; condition; variable update) {
    task to execute while for is true;
}
```
The for loop can be used to complete a task a set number of times, making repetitive tasks go quickly.

```c
while (condition is true) {
    task to execute while condition remains true;
}
```
The while loop is a good way to test for a certain condition to be met, making searching for something simple.

```c
do{
    task to execute;
} while (condition is true) ;
```
The do while loop is good for doing a task while some requirement remains valid.

Each of these loop types are useful for different tasks, making C useful for many different repetitive tasks. This highlights one of the main strengths of the language, being very fast.

I/O

C programming must be able to interface with a human user. This is accomplished by the input and output abilities. First is the output command. This could look like:

```c
printf("Hello, world!\n");
```
The printf command is the general command to print out to the user. The part in quotes is what will actually be displayed as text. This example would print out “Hello, World!” to the user. Notice the \n at the end. The \ means a spacing command in the text. \n means new line, and there are many others.

Sometimes a variable needs to be displayed to the user. This is done like shown here:

```c
printf("The answer is %i\n", answerVar);
```
The difference here is the %i and the variable name at the end. The %i is the spot where the variable should be output in the line. The %i means the variable is an integer, others, such as %f, will be
different variable types, such as float. After the end of the quote, a comma shows that a new command is coming. The answerVar shown is the variable name to be displayed where %i is in the output. For several variables to be shown in one command, each variable must be put inside of the quoted string, and then following the quotations, the variables can be typed out and separated by a comma in the order that they should appear in the output.

To get an input, the scanf command can be used. This looks like:

```c
scanf("%d", &a);
```

The order is very similar to the printf command. However, there is no text inside of the quotes. The %d, here a variable of type double is being taken. After the quotes, the &a means that the variable a will be assigned whatever value was input by scanf. Variable a must be of type double. Note a string input should look like this:

```c
scanf("%s", a);
```

The & should not appear before the variable name because the variable is an array, which do not use an address operator, which is what the & symbol is.

## Conclusion and References

The C language is very useful for many tasks, and this guide is hopefully a good refresher on how to do some simple tasks. The variables explained show a wide variety of abilities that a variable can have. The loops show some of the uses of the language and showcase some strengths of C that have made it withstand time. Finally, the I/O portion explains how the user can interact with a program.

- www.cprogramming.com
- en.wikibooks.org/wiki/C_Programming