CSE 113

Structured Programming Language

TASNIM ZAHAN
ASSISTANT PROFESSOR
DEPT. OF CSE
NEUB

Introduction

C programming is a general-purpose, procedural computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system.

- Procedural Language - Instructions in a C program are executed step by step.
- Portable - You can move C programs from one platform to another, and run it without any or minimal changes.
- Speed - C programming is faster than most programming languages like Java, Python, etc.
- General Purpose - C programming can be used to develop operating systems, embedded systems, databases, and so on.

Why Learn C Programming?

- C helps you to understand the internal architecture of a computer, how computer stores and retrieves information.
- After learning C, it will be much easier to learn other programming languages like Java, Python, etc.
- Opportunity to work on open source projects. Some of the largest open-source projects such as Linux kernel, Python interpreter, SQLite database, etc. are written in C programming.

Applications of C Programming

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C are -

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

Resources

- Text Book
  - Programming With C – Byron Gottfried (SCHAUM'S outlines)
  - Computer Programming – Tamim Shahriar Subeen (Bangla)
- Software
  - CodeBlocks: http://www.codeblocks.org/downloads/26
  - Install: codeblocks-20.03mingw-setup.exe

C Compiler

- Source code written in source file (text file) is the human readable
- It needs to be "compiled“ into machine language
- The most frequently used and free available compiler is the GNU C/C++ (gcc)compile

Hello World Example

```c
#include <stdio.h>

int main() {
   /* my first program in C */
   printf("Hello, World! \n");

   return 0;
}
```

Hello World Example

Let us take a look at the various parts of the above program −

- The first line of the program #include <stdio.h> is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation.
- The next line int main() is the main function where the program execution begins.
- The next line /*...*/ will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.
- The next line printf(...) is another function available in C which causes the message "Hello, World!" to be displayed on the screen.
- The next line return 0; terminates the main() function and returns the value 0.

Basic building blocks of C

- Identifier
  - 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).
- Keywords: predefined, reserved words used in programming that have special meanings to the compiler

  Examples: int, for, if, else, while, break ……
- Semicolon
  - In C semicolon is a statement terminator, that is, each individual statement must be ended with a semicolon

Basic building blocks of C

- Comment
  - Comments are like helping text in your C program and they are ignored by the compiler.

```
/* my first
   program in C */

// my first program in C
```

Data Types

- Data types in c used for declaring variables or functions of different types
- The type of a variable determines how much space it occupies in storage

Basic data types are listed below:

Data Type	Description	Size
int	Integer quantity	2 to 4 bytes (varies from one compiler to another)
long	Integer quantity	8 bytes
char	Single character	1 byte
float	Floating-point number	4 bytes
double	Double-precision floating-point number	8 bytes

Data type value range

- Integer variable: 2 bytes (16 bit)

  total combination: $2^{16} = 65536$ (-32768 to 32,767)

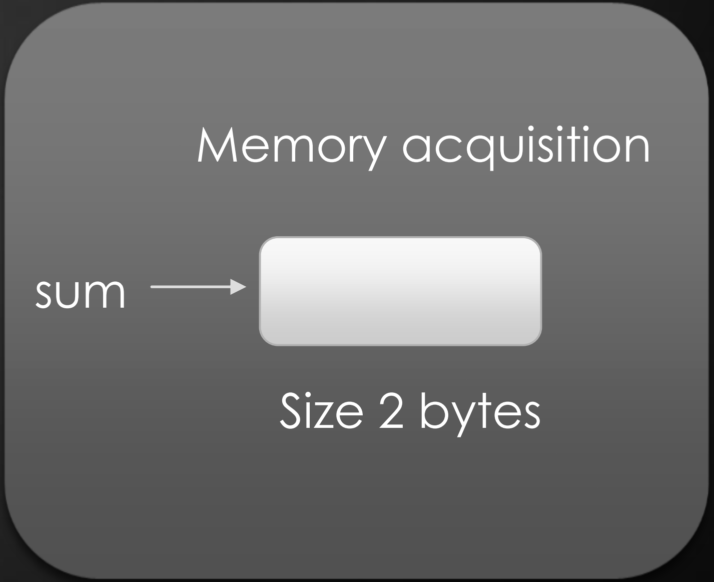
- Character variable: 1 byte (8 bit)

  total combination: $2^{8} = 256$ (ASCII characters)

Variable

- A variable is a name given to a storage area that our programs can manipulate
- Variable declaration

```
data_type variable_name;    ←——— Syntax
int sum;
float x,y;
char ch,a;
```



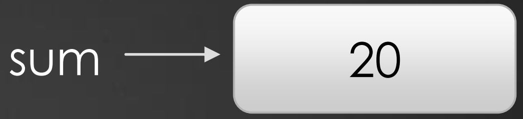
Memory acquisition

sum →

Size 2 bytes

Variable initialization

```
int sum=20;
float x=1.5,y=2.0;
char ch=‘A’;
```



sum

20

Size 2 bytes

Constants

- Integer constant: 12, 0, 999
- Floating-point constant: 1.34, 0.004e^-3
- Character constant: ‘A’, ‘b’, ‘9’
- String constant: “red”

Constants

#define identifier value

```
#include <stdio.h>

#define HIGHT 10
#define WIDTH  5
#define NEWLINE '\n'

int main() {
   int area;

   area = HIGHT * WIDTH;
   printf(“Area: %d", area);
   printf("%c", NEWLINE);

   return 0;
}
```

- ASCII (American Standard Code for Information Interchange)
  - 7 bit
  - Extended ASCII (8 bit)
  - A-Z (65 - 90),   a-z (97 - 122), 0-9 (48 - 57)

- UNICODE (16 bit)
- Escape Sequence: \n, \0, \’