BIT Notes

Chapter 2

By Karna Bahadur Shrestha

Character Set

The set of characters that are used to form words, numbers and expression in C is called C character set. The characters in C are grouped into the following categories.

- Letters or alphabets(a-z A-Z)
- Digits(0-9)
- Whitespaces
    - Blank spaces
    - Horizontal tab
    - Vertical tab
    - Carriage return

Special Characters

,	Comma	**&**	Ampersand
.	Period	^	Caret
;	Semicolon	*****	Asterisk
:	Colon	**-**	Minus Sign
?	Question Mark	+	Plus Sign
'	Aphostrophe	<	Opening Angle (Less than sign)
"	Quotation Marks	>	Closing Angle (Greater than sign)
!	Exclaimation Mark	**(**	Left Parenthesis
|	Vertical Bar	**)**	Right Parenthesis
/	Slash	**[**	Left Bracket
\	Backslash	**]**	Right Bracket
~	Tilde	**{**	Left Brace
-	Underscore	**}**	Right Bracket
$	Dollar Sign	#	.Number Sign
%	Percentage Sign		

Keywords and Identifiers

Every word in C language is a keyword or an identifier. Keywords in C language cannot be used as a variable name. They are specifically used by the compiler for its own purpose and they serve as building blocks of a c program.

The following are the Keyword set of C language.

auto	else	register	union
break	enum	return	unsigned
case	extern	short	void
char	float	signed	volatile
const	for	size of	while

continue	goto	static
default	if	struct
do	int	switch
double	long	typedef

Some compilers may have additional keywords listed in C manual.

Identifier refers to the name of user-defined variables, array and functions. A variable should be essentially a sequence of letters and or digits and the variable name should begin with a character.

Both uppercase and lowercase letters are permitted. The underscore character is also permitted in identifiers.

The identifiers must conform to the following rules.

1. First character must be an alphabet (or underscore)
2. Identifier names must consists of only letters, digits and underscore.
3. An identifier name should have less than 31 characters.
4. Any standard C language keyword cannot be used as a variable name.
5. An identifier should not contain a space.

Constants/Literals

A constant value is the one which does not change during the execution of a program. C supports several types of constants.

1. Integer Constants
2. Real Constants
3. Single Character Constants
4. String Constants

Integer Constants

An integer constant is a sequence of digits. There are 3 types of integer‟s namely decimal integer, octal integers and hexadecimal integer.

Decimal Integers: consists of a set of digits 0 to 9 preceded by an optional + or - sign. Spaces, commas and non digit characters are not permitted between digits. Examples for valid decimal integer constants are:

123
-31
0
562321
+78

Some examples for invalid integer constants are:

15 750

20,00
0 Rs.
1000

Octal Integers: constant consists of any combination of digits from 0 through 7 with an O at the beginning. Some examples of octal integers are:

O26
O
O347
O676

Hexadecimal integer: constant is preceded by OX or Ox, they may contain alphabets from A to F or a to f. The alphabets A to F refer to 10 to 15 in decimal digits. Examples of valid hexadecimal integers are:

OX2
OX8C
OXbcd
Ox

Real Constants

Real Constants consists of a **fractional** part in their representation. Integer constants are inadequate to represent quantities that vary continuously. These quantities are represented by numbers containing fractional parts like 26.082. Examples of real constants are:

0.0026
-0.97
435.29
+487.0

Real Numbers can also be represented by **exponential notation**. The general form for exponential notation is mantissa exponent. The mantissa is either a real number expressed in decimal notation or an integer. The exponent is an integer number with an optional plus or minus sign.

+3.2e5 [$+3.2x10^5$]

4.1e8 [$4.1x10^8$]

A Single Character constant represent a single character which is enclosed in a pair of quotation symbols.

Example for character constants are:

'5'
'x'
';'
' '

All character constants have an equivalent integer value which is called ASCII Values.

String Constants

A string constant is a set of characters enclosed in double quotation marks. The characters in a string constant sequence may be an alphabet, number, special character and blank space. Example of string constants are

"BSCCSIT"
"1234"
"God Bless"
"!.....?"

Backslash Character Constants [Escape Sequences]

Backslash character constants are special characters used in output functions. Although they contain two characters they represent only one character. Given below is the table of escape sequence and their meanings.

Constant	Meaning
'\a'	Audible Alert (Bell)
'\b'	Backspace
'\f'	Formfeed
'\n'	New Line
'\r'	Carriage Return
'\t'	Horizontal tab
'\v'	Vertical Tab
'\''	Single Quote
'\"'	Double Quote
'\?'	Question Mark
'\\'	Back Slash
'\0'	Null

Variables

A variable is a value that can change any time. It is a memory location used to store a data value. A variable name should be carefully chosen by the programmer so that its use is reflected in a useful way in the entire program. Variable names are case sensitive. Examples of variable names are

Sun
number
Salary
Emp_name
average1

Any variable declared in a program should confirm to the following:

1. They must always begin with a letter, although some systems permit underscore as the first character.
2. The length of a variable must not be more than 8 characters.
3. White space is not allowed and
4. A variable should not be a Keyword
5. It should not contain any special characters.

Examples of Invalid Variable names are:
123
(area)
6th
%abc

Data Types in C

A C language programmer has to tell the system before-hand, the type of numbers or characters he is using in his program. These are data types. There are many data types in C language. A C programmer has to use appropriate data type as per his requirement in the program he is going to do.
There are basic three categories:

- Primary Data types
- User defined Data types
- Derived Data types

Primary data type

All C Compilers accept the following fundamental data types

1.	Integer	int
2.	Character	char
3.	Floating Point	float
4.	Double precision floating point	double
5.	Void	void

Integer Type

Integers are whole numbers with a machine dependent range of values. A good programming language as to support the programmer by giving a control on a range of numbers and storage space. C has 3 classes of integer storage namely short int, int and long int. All of these data types have signed and unsigned forms. A short int requires half the space than normal integer values. Unsigned numbers are always positive and consume all the bits for the magnitude of the number. The long and unsigned integers are used to declare a longer range of values.

Floating Point Types

Floating point number represents a real number with 6 digits precision. Floating point numbers are denoted by the keyword float. When the accuracy of the floating point number is insufficient, we can use the double to define the number. The double is same as float but with

longer precision. To extend the precision further we can use long double which consumes 80 bits of memory space.

Void Type

Using void data type, we can specify the type of a function. It is a good practice to avoid functions that does not return any values to the calling function.

Character Type

A single character can be defined as a defined as a character type of data. Characters are usually stored in 8 bits of internal storage. The qualifier signed or unsigned can be explicitly applied to char. While unsigned characters have values between 0 and 255, signed characters have values from –128 to 127.

Size and Range of Data Types on 16 bit machine;

TYPE	SIZE (bytes)	Range	Format Specifier
Char or Signed Char	1	-128 to 127	%c
Unsigned Char	1	0 to 255	%c
Int or Signed int	2	-32768 to 32767	%d
Unsigned int	2	0 to 65535	%u
Short int or Signed short int	2	-128 to 127	%d or %i
Unsigned short int	2	0 to 255	%u
Long int or signed long int	4	-2147483648 to 2147483647	%ld
Unsigned long int	4	0 to 4294967295	%lu
Float	4	3.4 e-38 to 3.4 e+38	%f
Double	8	1.7e-308 to 1.7e+308	%lf
Long Double	10	3.4 e-4932 to 3.4 e+4932	%Lf

Declaration of Variables

Every variable used in the program should be declared to the compiler. The declaration does two things.

1. Tells the compiler the variables name.
2. Specifies what type of data the variable will hold.

The general format of any declaration

data-type variable_name1,variable_name2,……..,variable_name_n;

Where *variable_name1*, *variable_name*2, *variable_name3* are variable names. Variables are separated by commas. A declaration statement must end with a semicolon.

Example:

int sum;
int number, salary;
double average, mean;

Datatype	Keyword Equivalent
Character	char
Unsigned Character	unsigned char
Signed Character	signed char
Signed Integer	signed int (or) int
Signed Short Integer	signed short int (or) short int (or) short
Signed Long Integer	signed long int (or) long int (or) long
UnSigned Integer	unsigned int (or) unsigned
UnSigned Short Integer	unsigned short int (or) unsigned short
UnSigned Long Integer	unsigned long int (or) unsigned long
Floating Point	float
Double Precision Floating Point	double
Extended Double Precision Floating Point	long double

User defined type declaration

In C language a user can define an identifier that represents an existing data type. The user defined datatype identifier can later be used to declare variables. The general syntax is

typedef type identifier;

Here type represents existing data type and „identifier" refers to the „row" name given to the data type.

typedef int integer;
Here integer symbolizes int data type, Now we can declare int variable “a ” as *integer a* instead if *int a* (but *int a)* is still valid.

Derived datatypes

There are some datatypes which are derived from the existing primary data types.

struct st{
int a;
float b;
char c;
}

Delimeters

A **delimiter** is one or more characters that separate text strings. Common **delimiters** are commas (,), semicolon (;), quotes (", '), braces ({}), pipes (|), or slashes (/ \ ).

Expressions

An expression in C is defined as 2 or more operands are connected by one operator and which can also be said to a formula to perform any operation. An operand is a function reference, an array element, a variable, or any constant. An operator is symbols like “+”, “-“, “/”, “*” etc.

Let’s observe this example:

A*B

In the above expression multiplication symbol (*) is said to be an operator and A and B are said to be 2 operands.

Comments

Comments in C language are used to provide information about lines of code. It is widely used for documenting code. There are 2 types of comments in the C language.

1. Single Line Comments
2. Multi-Line Comments

Single line comments are represented by double slash //. Let's see an example of a single line comment in C.

Multi-Line comments are represented by slash asterisk * ... *\. It can occupy many lines of code, but it can't be nested. Syntax:

```
/*
code
to be commented
*/
```

C tokens

The basic elements recognized by the C compiler are the tokens. A token in source program text that the compiler does not breakdown into component elements. The keywords, identifiers,