Unit 1 Introduction to C Language

Structure:

1.1 Introduction

C is a general-purpose, block structured, procedural, imperative computer programming language developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories. Every program is limited by the language which is used to write it. C is a programmer's language. Unlike BASIC or Pascal, C was not written as a teaching aid, but as an implementation language. C is a computer language and a programming tool which has grown popular because programmers like it! It is a tricky language but a masterful one. If you have come to C in the hope of finding a powerful language for writing everyday computer programs, then you will not be disappointed. C is ideally suited to modern computers and modern programming.

Objectives:

At the end of this unit, you will be able to:

- define variables and constants in C Language
- explain different Data types and their sizes in C Language
- construct expressions in C Language

1.2 Historical Development of C language

Before 1960, there was no such language by which all types of programming can be done. There was BASIC (**B**eginners **A**ll-purpose **S**ymbolic **I**nstruction **C**ode) for graphical application, COBOL (**C**ommon **B**usiness **O**riented **L**anguage) for business operation, PASCAL for scientific application, FORTRAN (**FOR**mula **TRAN**slation) numerical and scientific applications etc. Then the programmer faced problems about that and a Committee was formed for a language so that it can able to solve all kinds of problems. In late 1960s, the Committee came out with a language called ALGOL-60. But the language was too complicated to write a simple program and that was not approved. After a long period of time, in early 1970s, the Committee came out with a language called BCPL (Basic Combine Programming Language) or Language B. After that in 1972 the **Dennis Ritchie** published a language from AT & T Bell Laboratory called **C language**. After that it was still being developed for advance languages.

C language is the mother language of all other programming languages. The main thing is that, here is the wide area for build up logic. All other languages are developed from the base of C language. Like C++, Java, C#, Script Languages etc. So it is very important to realize the language so that the knowledge can be made applicable to all the other languages. The C language is 3rd Generation Language (3GL) but it sometimes called min-level language because it also supports the bit level (log level) programming.

1.3 Character Sets

Like other natural languages [English, Hindi, Bengali etc.] we need to know about the language character sets. In English, we first have to know the alphabets. Like this, C language has 98 character sets, which are given below:

Characters	Name	Count
A – Z, a-z	Letters	52
0-9	Digits	10
, ; : ? ‘ “ ! @ $ # % ^ | . / \ { } [ ] ( ) * + - _ & = ~ > <	Special Characters	31
Blank space, horizontal tab, Carriage return, New line, From feed	White Spaces	5
Total		**98**

Table 1.1: C language Character Set

1.4 Variables

Variables are the names given to locations in the memory or CPU register of computer where different constants (Values) are stored. The variable name may take different values at different times during executions of programs.

Programming Language sometimes is analogous to the cooking operations. For cooking, we need some utensils like plates, cups, glasses, jugs, large containers etc., in programming we need some variables to store data. There are some naming conventions to declare the variable.

Rules for constructing variable names:

Any sequence of characters from the ‘C’ characters set confirming the following rules form a valid variable name.

1) The letters, the digits and the special character (_) underscore can only be used.

   Ex.: student_age, college_name, student1, class_1 etc.

2) The variable name must begin with a letter but some compiler permits underscore as the first character.

Ex.: Office_1, India1, mobile

3) There are almost 40 characters.

Ex.: school_college_office_city_village_country_name [Invalid]

4) The variable name is case sensitive.

Ex.: Student_Name and **student_name** are not same

5) There must not be any blank.

Ex.: Country Name [Invalid]

6) The variable should not be any keyword.

Ex: for, void, int etc.

1.5 Keywords

Keywords are the set of words in C language meaning and actions of which remains fixed throughout the language. We cannot use the following 32 keywords as variable name.

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

Table 1.2: Keywords in C Language

1.6 Data Types

Data type is essential when we define a variable. Because when variable is defined we must maintain the data type along with the variable so that it possible to allocate the memory space.

Let us consider the analogous example of cooking. At the time of defining a container we must define its capacity. If it is large enough, then space is wasted and if is small then data can be overflowed.

In C language, there are four basic data types.

a) Primary or Fundamental Data Type

b) User Defined Data Type

c) Derived Data Type

d) Empty Data Type

a) Primary or Fundamental Data Types:

The primary data types are built in and with the help of that data type we can create all other data types. The Primary or Fundamental Data Types are again divided into three types as given below:

Data type	Range of Values	Memory Occupies (Bytes)	Scan Format	Code
Integer				
Sign Sort Integer	-128 to +127	2	%d	short int
Unsigned short	0 to 255	2	%d	unsigned short
Sign Integer	-32768 to + +32768	2	%d	int
Unsigned integer	0 to 65,535	2	%u	unsigned int
Sign Long Integer	-2147483648 to – 2147483648	4	%ld	long int
Unsigned long	0 to 4294967295	4	%lu	unsigned long
Floating Point				
Floating Point	$-3.4X10^{38}$ to $-3.4X10^{38}$	4	%f	float
Double	$-1.7X10^{308}$ to $1.7X10^{308}$	8	%lf	double
Long double	$-1.7X10^{4932}$ to $1.7X10^{4932}$	10	%lf	long double
Character				
Signed character	-128 to 127	1	%c	char
Unsigned character	0 to 255	1	%c	unsigned char

Table 1.3: Primary Data Types

b) User Defined Data Types

The user defined data types are defined by using primary data types.

Example: Structure, Union, Enum, Array etc.

c) Derived Data Types

The derived data types are the types that are derived from fundamental data types.

Example: Arrays, Pointers etc.

d) Empty Data Types

The empty data type means that it can be assigned by any other data type.

Example: void

1.7 Constants

The constants are the fixed values that are stored to the variable which cannot be altered. In C language there are two types of constants:

i) Primary Constant ii) Secondary Constant

i) Primary Constant: The constant, which contains only one data type called primary constant. The primary constants are

1) **Real Constant:** All the numbers from real number system are called Real Constants.
   Example: 34, 74.905, -73.09, 782 etc.
2) **Single Character Constant:** This type of constant contains only one character within single quote. The ASCII comparison is possible with these constants.
   Example: 'A', '4', '=' etc.
3) **String Constant:** This type of constant is the set of combination of characters enclosed with double quote. The characters may be letters, digits, special characters or white spaces.

4) **Backslash Character Constants:** Language 'C' supports some special backslash character (control characters) constants that are used in output functions

CONSTANTS	MEANING	CONSTANTS	MEANING
'\a'	Audio able alert (bell)	'\v'	Vertical tab
'\b'	Back space	'\' '	Single quote
'\f '	Form feed	'\" '	Double quote
'\n'	New line	'\?'	Question mark
'\r'	Carriage Return	'\\'	Back slash
'\t'	Horizontal tab	'\⊘'	NULL

Table 1.4: Control Characters

ii) Secondary Constant: The constant, which contains more than one data type is called secondary constant.

Example: Array, Pointer, Structure, Union, Enum etc

1.8 Operators and Expressions

To write C language statements we need several operators and expressions. The operators have some special meaning and syntax rather grammar, so we should follow that to build any C language statement just as in English Language, there are some grammar aspects that we must know to construct correct sentences. There are EIGHT types of operators.

A) Arithmetic Operator: These types of operators are used especially for arithmetic operations. The operators are given below:

Operators	Meaning	Operators	Meaning
+	Addition	/	Division
-	Subtraction	%	Modulo Division
*	Multiplication	()	Brackets

Table 1.5: Arithmetic Operators

B) Relational Operator: It creates a relation between two expressions.

Operators	Meaning
<	Is less than
<=	Is less than or equal to
>	Is greater than
>=	Is greater than or equal to
==	Is equal to
!=	Is not equal to

Table 1.6: Relational Operators

C) Logical Operator: This operator creates logical relations between two expressions.

Operators	Meaning
&&	Logical AND
||	Logical OR
|	Logical NOT

Table 1.7: Logical Operators

D) Assignment Operator: This type of operator is used to assign the result of an expression to a variable. We have seen the usual assignment operator "=". In addition, C language supports a set of short hand assignment operators of the form.

Without Short Hand Operators	With Short Hand Operators
a = a+1	a + = 1
a = a-1	a - =1
a = a*(n+1)	a*=n+1
a = a/(n+1)	a/=n+1
a = a%b	a%=b

Table 1.8: Assignment Operators

E) Increment & Decrement Operators: In C language, the important operators are found which are generally not found in other languages. These operators are specially used for increasing or decreasing the value of the variable by one only. The operators are given below:

Without Increment Operators	With Increment Operators
a = a+1	a ++ or ++a
a = a-1	a - - or - - a

Table 1.9: Increment and Decrement Operators

Note: ++a (Pre Increment) and a++ (Post Increment)are not sometimes the same

++a => First increment the value of 'a' by 1 and then compare.

a++ => First compare and then increment the value of 'a' by 1.

Example: If initially b = 40

i) a=b++; Then after execution of the expression the value of a=40 and b=41

ii) a=++b; Then after execution of the expression the value of a=41 and v=41

F) Conditional Operator: A ternary operator pair "**?:**" is available in language 'C' so that the simple conditions can take place simply.

<Condition>? <Exp. for true value>: <Exp. for false value> ;

Example : Initially a=10, b=20 ,c=30	
I) (a>7)?b=b+5:c=c-5; After execution the value of : a=10 b=25 c=30	(a>70)?b=b+5:c=c-5; After execution the value of : a=10 b=20 c=25

G) Bitwise Operator: One of the C language's powerful features is a set of bit manipulation operators. These permit the programmer to access and manipulate individual bits within a piece of data. The various bitwise operators available in C language are the following:

Operators	Meaning
~	One's complement
>>	Right shift
<<	Left shift
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR

Table 1.10: Bitwise Operators

H) Special Operator: The language 'C' supports some special operators such as Comma Operator (,), sizeof operator, pointer operator (& and *) and member section operator (• And →). There are also type processor operators known as "**String-izing**" and "**Token pasting**" operators (# and ##) available in C.

1.9 Summary

C is a general-purpose, block structured, procedural, and imperative computer programming language developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories. C is a programmer's language. Unlike BASIC or Pascal, C was not written as a teaching aid, but as an implementation language. C is ideally suited to modern computers and modern programming. The historical development and the basic units of C language such as character sets, variables, data types, operators and expressions focused in this unit.

Self Assessment Questions

1. Dennis Ritchie published language C from AT & T Bell Laboratory in the year of _____ .
2. There are _____ numbers of characters in C Language.
3. ____ number of Keywords in Language C.

1.10 Terminal Questions

1. How many characters in C language?
2. What is data type? Why it is required?
3. What is the difference between pre increment and post increment operator?
4. Can we increment the value of a variable by 2 using any increment operator?
5. What is the difference between string type and character type constants?
6. What is the purpose of bitwise operators?

1.11 Answers to Self Assessment and Terminal Questions

Self Assessment Questions

1. 1972
2. 98
3. 32

Terminal Questions

1. Section 1.3
2. Section 1.6
3. Section 1.8 – E
4. Section 1.8 – D
5. Section 1.7
6. Section 1.8 – G