

Lecture Handout - 2**1) INTRODUCTION TO C**

- C is a popular **general purpose programming language**.
- C language has been **designed and developed** by **Dennis Ritchie** at **Bell Laboratories** in **1972**.
- It is an offspring of the “Basic Combined Programming Language” called ‘B’ .
- C Programs are **efficient, fast and portable**.
- C language is a **middle level Computer Language**.
- It combines the features of high-level language and the functionality of assembly language.
- It is well suited for writing both application software and system software.
- **C is a structured language.**

2) Structure of a C program

- **Every C program contains** a number of several **building blocks known as functions**.
- Each function performs task independently.
- A C program comprises of the following sections.

Include Header File Section
Global Declaration
/* Comments */
main() Function name
{
/* Comments */
Declaration Part
Executable Part
}
User-defined function
{
}

Example :
/* This is my First Program */
#include <stdio.h>
main()
{
printf(“ Hello World”);
}

a) Include Header File Section:

- Every C program must include some **header files for function definition**.
- Each **header file** by default is extended with **.h**.
- The file should be included using **#include** directive.



- **Example:** `#include <stdio.h>` or `#include "stdio.h"` – for standard input & output

b) Global Declaration

- The **variable declared** in this section can be used **by more than one function**.
- These variables are called **global variables**.
- They are declared outside of all the functions.

c) Main Function

- Every program written in C language must contain **main() function**.
- The **execution of C program** always begins with the function main().
- The Program starts with the **opening curly brace{** and ends with the **closing brace}**.
- Between these two braces the programmer should declare the declaration and executable part.

d) Declaration Part

- The declaration part **declares the entire variables** that are **used in executable part**.
- The **initialization for variable** is done here.
- The **initialization** means providing **initial (starting)** value to the variables.

e) Executable Part

- This contains **set of statements** or a **single statement**.
- These are **executable statements** followed by the declaration of the variables.
- These statements are **enclosed between braces**.

f) User-defined Function

- **The functions defined by the user are called user-defined functions.**
- These functions are defined after the main() function.
- User-defined function **is optional**.

g) Comments

- **Comment statements are not the part of executable program.**
- To understand the **flow of the programs** the programmer can include comments in the program.
- **Comments** are the statements **placed between** the delimiters **/* comments */**.
- **The compiler does not execute comments.**

Programming Rules: Every program should abide the following rules:

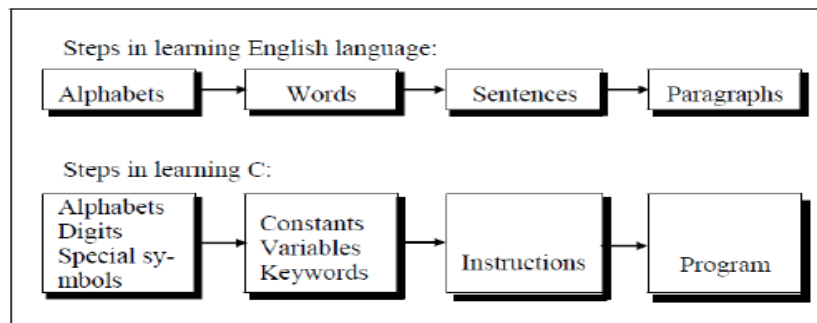
1. All statements should be written in lower case letters. C is a case-sensitive language.
2. Upper case letters are only used for symbolic constants.
3. Blank spaces may be inserted between the words to improve readability.



4. The programmer can write the statement anywhere between the two braces following the declaration part. The user can write one or more statements separating them by a semi-colon(;). It is a free-form language.
5. The opening and closing braces should be balanced.

3) C CHARACTER SET - INTRODUCTION

- A program is a **set of statements** for a **specific task**.
- It will be executed in a **sequential form**.
- These statements (or) instructions are formed using certain **words and symbols** according to the rules known as **syntax rules** or **grammar** of the language.
- Every program must follow accurately the syntax rules supported by the language.



C Character Set

The characters used to form words, numbers and expressions depend upon the computer on which the program runs.

The characters in C are classified in the following categories:

1. Letters
2. Digits
3. White Spaces
4. Special Characters

1) Letters	2) Digits	3) White Spaces
A,B,C,D,E,F,G,H,I,J,K,L,M,N, O,P,Q,R,S,T,U,V,W,X,Y,Z	0 - 9	Blank space (Space bar)
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r ,s,t,u,v,w,x,y,z		Horizontal tab (\t)
		Vertical tab (\v)
		New line (\n)
		Form Feed (\f)



Special Characters

,	Comma	&	Ampersand
.	Period or Dot	^	Caret (Cap)
;	Semicolon	*	Asterisk
:	Colon	-	Minus
`	Apostrophe	+	Plus
“	Quotation Mark (Double Quotes)	<	Less tag (Open tag)
!	Exclamation Mark	>	Greater than (Close tag)
	Vertical Bar	()	Parenthesis left/right
/	Slash	[]	Bracket left / right
\	Back Slash	{ }	Braces left / right
~	Tilde	%	Percentage
_	Under Score	#	Hash or Number Sign
\$	Dollar	=	Equal to
?	Question Mark	@	At the rate

Delimiters

Language pattern of C uses special kind of symbols as separators called as **delimiters**. They are as follow:

Delimiters	Use
: Colon	Used for Label
; Semicolon	Terminates Statement
() Parenthesis	Used in expression and Function
[] Square Bracket	Used for array declaration
{ } Curly Braces	Scope of statements
# Hash	Preprocessor Directive
, Comma	Variable Separator



4) C Keywords

- The C Keywords are reserved words by the Compiler.
- All the C keywords have been assigned fixed meaning.
- These keywords cannot be used as variable names because they have already assigned fixed jobs.
- They are follows:

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

Identifiers

- Identifiers are the names of variables, functions and arrays.
- They are user defined names.
- They may contain sequence of letters and digits, with letter as the first character.
- Lower case letters are preferred. (Upper case are allowed)
- In special character, only under score (_) is used.

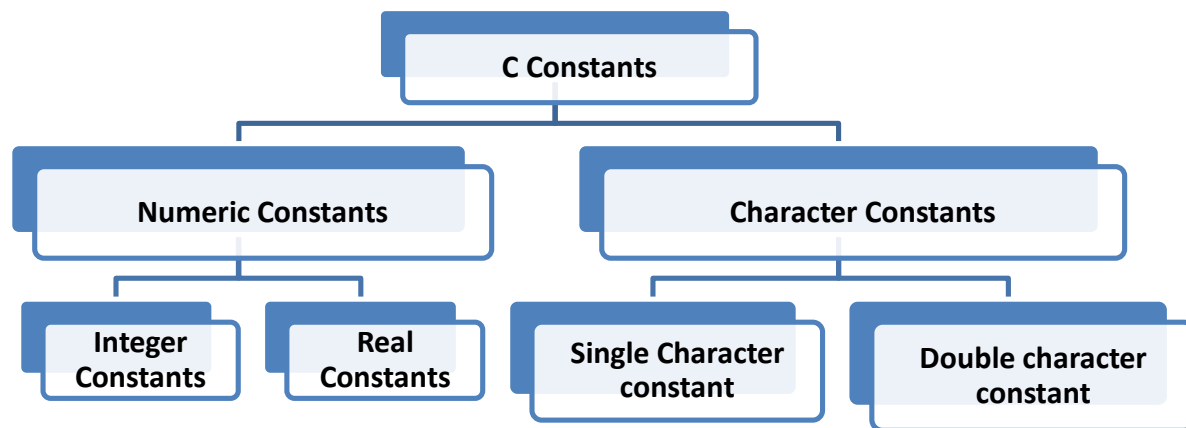
Example:

```
#define N 10      int i;  
#define a 15      char c;
```

C Constants

- The C Constants are the values that do not change during the execution of a program.
- There are several types of constants in C.





A. Numeric Constants

1. Integer Constants:

- These are the **sequence of numbers** from **0 to 9** without **decimal points** or fractional part or any other symbols.
- It requires **minimum two bytes** and **maximum four bytes**
- Integer constants could be **positive** or **negative** or **zero**.
- The number **without sign** is assumed as **positive**.
- **Example:** 10, 20, +30, -15

2. Real Constants

- **Real Constants** are often known as **Floating point** constants.
- The Real constants are written in exponential notation, which contains a fractional part and an exponential part.

General format contains: **Mantissa** and an **Exponent**.

Mantissa: A real number represented in decimal or an integer.

Exponent: A exponent is an integer number may be +ve or -ve.

The letter '**e**' (**lower case**) **separates** the mantissa and exponent.

Example: Length, height, prize, distance are measured in real numbers.

1) 2.5, 2) 5.521 3) 2456.123 can be written as $2.4561 \times e^{-3}$

B. Character Constants

1. Single Character Constants:

- A character constant is a single character.
- They are enclosed within a pair of single quote marks.



- They may represent a single special symbol or white space.
- Character constant have integer value known as ASCII values. Eg: `printf("%c %d", 65, 'B')` will display 'A',66.

Examples:

'a', '8', '@', ' '

2) String Constants

- **String Constants** are sequence of characters enclosed within double quote marks.
- The string may be a combination of all kinds of symbols.

Example:

"Hello World", "India", "My Roll No: 12345978"

"Hey @ 123..."

VARIABLES

- A variable is a data name used for storing s data value.
- Its value may be changed during the program execution.
- Eg: Height, average,sum.

Rules for Defining Variable:

1. They must begin with a character without spaces.
2. In special symbols only underscore is permitted.
3. The length of the variable should not exceed 8 characters.
4. Variable name should not be a keyword.
5. Variable name should not start with a digit.

Declaring Variables:

- Declaration of the variable should be done in the declaration part of the program.
- The variable must be declared before they are used in the program.
- Compiler allocates memory for the variable during declaration with the size of the data type specified.

Syntax: Data type variable;

Examples: int i;

char c; float f;

Syntax: variable = value;

Initializing Variables:

Initializing a variable can be done by the assignment operator "=".

Example: i = 10; or int i=10;



EXECUTING THE PROGRAM

The following steps are essential in executing a program in 'C'.

a) **CREATION**

- Programs should be written in C editor.
- The file name does not necessarily include extension C.
- The default extension is C.
- The user can also specify their own extension.

b) **COMPILATION AND LINKING OF A PROGRAM**

- The source program statements should be translated into object programs to be executed by the computer.
- The translation is done after correcting each statement.
- If there is no error the compilation proceeds, if errors the programmer should correct them.
- The translated program is stored in another file with the same name with extension “.obj”.
- Compiler links all other program files and function together that are required by the program.
- E.g 1: printf, scanf library functions in stdio.h are linked to the main() program.
- E.g 2: pow() library function in math.h is linked to the main() program.

c) **EXECUTING THE PROGRAM**

After the compilation the executable object code will be loaded in the computer's main memory and the program is executed.

.....

