BASIC C

1. In 1967, Martin Richard developed a language called BCPL (Basic Combined Programming Language) to develop only System Software
2. In 1970, Ken Thompson created a language using many features of BCPL and called it simply ‘B’ language
3. ‘B’ language was used to create early version of UNIX Operating System at AT&T Bell laboratory in USA
4. ‘C’ language Evolved from ALGOL, BCPL, B language by Dennis Ritchie at Bell Labs in 1972
5. In 1989 C language was approved as a Standard Language by ANSI (American National Standard Institute)

Characteristics

1. C is General Purpose Structured Oriented Programming Language
2. Used to develop both System and Application Software's
3. It is Middle Level Language
4. Fast when compared with other languages
5. Robust (Strong) Language whose rich set of built-in functions and operations can be used to write any complex program.
6. Portable Language : Program written in C language can be easily transferred form one computer to another computer
7. Used to develop software like
    a. Graphical Packages,
    b. Office Automation Packages,
    c. CAD/CAM Software
    d. Database Management Software and
    e. Scientific and Engineering Software

Structure of C Language

1. Documentation Section
2. Linking Section
3. Global Declaration
4. Main Section

{

```
                Local Declaration;
                Executable Part;
        }
    5.  Sub Functions
            (a) Function1()
            (b) Function2()
```

1. **Documentation Section**
    a. Used to Describe the program statements in a C file
    b. Given as a Comment Line

        /* Comment Line Sentence */

    c. It is optional
    d. It can be any where in the program.
    e. Compiler will not translate the Comment Line

2. **Linking Section**

    a. Which is used to link the pre-defined library functions into a C program

```
#include<stdio.h>
#include<math.h>
#include<graphics.h>
```

3. **Global Declaration**

Used for declaring the variables and constants, which are accessable by all the functions in a c program file.

Example:

const float pi= 3.14;

4. **Main Section**

It is used to start the execution of a program

```
void main()
{
        --------------;
```

```
            ---------------;
            ----------------;
        }  /*Compound Statement*/
```

Local Declaration

To declare memory allocations for constants, variables, arrays .. required by a program

Syntax

datatype variable1, varibale2,……;

Eg:

float radius, area;

Executable Part

This part is used to write the actual programming statements to execute. Those statements are run on a CPU

Eg:

Area = pi * radius * radius;

5. Sub Function

a. Functions are sub programs
b. A block of programming statements which are execute when called.
c. It may be call by many time (Reusable of Code)
d. Helps for Modular Programming

Chapter - 2

1. **C Character Set**
    1. Alphabets
        - A to Z
        - a to **Z**
    2. Number
        - 0 to 9
    3. Special Characters
        - + - * / % { } $ ….
2. **C Tokens**

   Tokens are the smallest individual units in a program

Type of Tokens:

(a) Keywords or Reserved Word

A word whose meaning has been defined already in the C Complier.

There are 32 keywords

Example :

int, float, double, break, if ,else,switch,while, do, …….

(b) Constant

A value that does not change during the execution of a program is called Constant

Syntax:

const data_type identifier=Constant_value;

Example

const float pi =3.14;

Constant

(c) Variables

A quantity or a value, which may vary during execution of program, is called as Variables

Syntax :

Datatype *variable1, varibale2,……;*

Example:

float radius,area;

(d)Identifiers

Identifiers are names used to identify various programming elements in a C program, such as

a. Variables,
b. Constants,
c. Arrays,
d. Functions,
e. Structures, Unions, labels etc.,

Rule for Naming Identifier

(a) The First Character must be start with alphabet or an underscore and can be followed only by any number of alphabets, or digits or underscores.
(b) Maximum of 256 characters length
(c) No other Special Characters except underscore
(d) Should not be a Keyword
(e) No Space between Identifier
(f) Upper Case Identifiers is distinct from Lower Case Identifier name.
Eg: SALARY is not equal to salary

Valid Identifier

(a) a,b,c;
(b) salary,basic_pay;
(c) A1;
(d) A_1;
(e) Abc,xyz,pi

Invalid Identifier

(a) break; (keyword)
(b) 1a; (Should not start with number)
(c) basic pay; (No Space)
(d) abc$; (Sp. Character)

(e) Data Types

It represents type of values processed by the programming language.

Primary Data Type

Data Types and Range

3. Formatted Standard Input & Output Statement

(a) Input Statement (Keyboard)

Syntax:

```
Scanf(“format”,&varibale1,&varibale2,…);
```

Example:

```
int a;
float f;
scanf(“%d%f”,&a,&f);
```

& - address of a variable

(b) Output Statement (Monitor)

Syntax:

```
printf(“Format + constant String”,var1,var2……);
```

Example:

```
printf(“%d”,a);
          (or)
printf(“A = %d”,a);
          (or)
printf(“Welcome to C programming “);
```

4. Escape Sequency (or) Back Slash Constant

\n	– New Line
\a	- one Beep Sound
\t	- Tab Space (5 Space)
\r	- Carriage Return (print on same line)
\b	- Back Space(Erase one Char.)
\”	- print Double Quote

Steps to Develop a C program

a.	Start -> Run -> Type cmd or command ->click Ok	
b.	To maximize the window	alt+enter
c.	to change mode	mode con lines=25
d.	Select the Drive	d: or e: or f:
e.	Make a Directory	md <name>
f.	Change a Directory	cd <name>
g.	To open a C editor	tc
h.	To open a New File	alt+f,n
i.	To Save a File	press f2
j.	To compile a File	Alt + f9
k.	To Run a Compiled file	Ctrl+f9
l.	To maximize the window	f5
m.	Go to next program	f6
n.	Step by step execution	f7
o.	To close a current program	Alt+f3
p.	To close all the programs	Alt+w+e

Execution of C Program

Chapter 3

1. Operators

Types of Operators

- **a. Unary Operator**

  Operators that acts upon a single operand

- **b. Binary Operator**

  Acts upon two operand

- **c. Ternary Operator**

  Acts upon three operand

Operators

- a. Arithmetic Operator (Binary)
- b. Relational Operator (Binary)
- c. Logical Operator (Binary)
- d. Conditional Operator (Ternary)
- e. Assignment Operator (Binary)
- f. Increment / Decrement Operator (Unary)
- g. Comma Operator (Binary)
- h. Sizeof Operator (Unary)
- i. Bit-Wise Operator (Binary)

a. Arithmetic Operator - Binary

i. Expression

Combination of Operand(variable or constant) and Operator(+.-.*,/,%..)

Eg: a+b;

a+b/c;

ii. Operator Priority

- a. Left to Right
- b. ( )
- c. ++,-- (Right to Left)
- d. *,/,%
- e. +,-

iii. Type Casting

Converting from one data type to another data type

- **a. Implicit Type Casting (Complier)**
- b. **Explicit Type Casting (Programmer)**

a. **Implicit Type Casting**

b. **Explicit Type Casting**

Syntax:

```
Variable = (typecast) expression;
```

Example:

```
int a,b,c;
float f;
f= (float) (a+b)/c;
```

2. **Relational Operator**

Which is used to make decision for real world problem

The expression will give either True or False

True =1

False=0

Relational Operators

3. **Logical Operators**

It is used to evaluate two relational expressions

Which also used to make a decision either true or false

Logical Operator

4. **Conditional Operator**

The operator that act upon three operands are called as Conditional Operator or Ternary Operator.

This operator is used to execute one only true or false statement.

Which is used to make a decision

Syntax:

```
Exp1 ? Exp2 : Exp3;
```

If Exp1 is true, Exp2 will execute

If Exp1 is false , Exp3 will execute

Example

```
{
int a=5,b=3,big;
big=(a>b) ? a : b ;
printf("Biggest of two nos = %d",big);
}
```

Output:

Biggest of two nos = 5

5. **Assignment Operator (=)**

The operator that is used to assign the result of an expression to a variable is called as an Assignment Operator.

Types of Assignment Operator

Post Assignment

```
a=a+b;
```

Pre Assignment

```
a+=b;
```

6. **Increment(++)/Decrement (--) Operator**

Increment by 1 or Decrement by 1 with current value.

Types

a. Post Increment and Decrement
b. Pre Increment and Decrement

7. **Comma Operator(,)**

Which is used to link the related Expressions together

Example:

```
int a,b,c;
c=(a=10,b=20,a+b);
Printf("Sum of two nos = %d",c);
```

8. **Sizeof Operator**

Used to get the size of a data type,Variable or Constant in bytes

This operator will return integer

Example:	**Output:**
Sizeof(int);	2
Sizeof(float);	4
Sizeof(long double);	10

9. **Bitwise Operators**

The operators which are used to manipulate the data in terms of bits.

Eg: int a=5, b=3