Fundamentals of "C" language

1:- What is c language?

Ans:- C is a computer programming language. C language replaced traditional programming languages of that time like: FORTRAN, PL/I and ALGOL. C was developed in 19?? at AT and T's bell laboratories of USA, "Dennis Ritchie" is known as designer and writer of C language. C successfully combines the structure of HLL and the power and efficiency of assembly language or low level language.

2:- Features of c language:-

1:- C has a relatively easier syntax than FORTRAN, COBOL and PASCAL.

2:- C is an efficient and fast programming language. It is more than 50 times faster than BASIC.

3:- C language contains vast set of data types as compared to languages of its times.

4:- C language has 32 most commonly used keywords. In addition, 8 keywords are used with system programming.

5:- C language is a highly portable language. This means, if you have created a C program on one computer. It will run on the other computers without any modifications. The only exception may be a very different type of operating system that is not compatible with older OS in which program has

been created.

6.) C is a very well suited language for structured programming. The programmer can easily divide a problem into a number of modules or functions.

7.) C has got a rich set of library functions. These are the functions that provide ready-made functionality to users.

8.) C also has a support for graphics programming.

9.) C is an extendible language. This means users can add their own functions to the library set.

10.) Various versions of C language are available from various companies like Borland C Turbo C and ANSI C

11.) C can efficiently deal with bit, byte and word addresses.

3.) Programming Techniques :-

Programming in C is very similar to learning a language like hindi or English. We can understand the topic by below figures :-

Learning English

Alphabets ⇒ Words ⇒ Sentences ⇒ Paragraphs

Learning C programming :-

Alphabets, Digit, Symbols	⟹	Constants Variables Keywords	⟹	Instructions	⟹	Program

—— Learning C language ——

4.1 Character Set :-

A character can be any alphabet, digit or special Symbol Used to represent a piece of Information

Alphabets :- A B C ..... YZ
a b c ..... yz

Digits :- 0, 1, 2, 3 ..... 8, 9

Special Symbols :- , — ' @ # % ^ & * ( ) _ - + = / \ | { } [ ] ; : " ' < > . ? !

Here, Commas are used to separate two symbols and it is also treated as a valid special.

1.1 Alphabets :-

C Character Set supports alphabets from A to Z and a-z. Note that C is case sensitive language which means capital (A-Z) is different than (a-z)

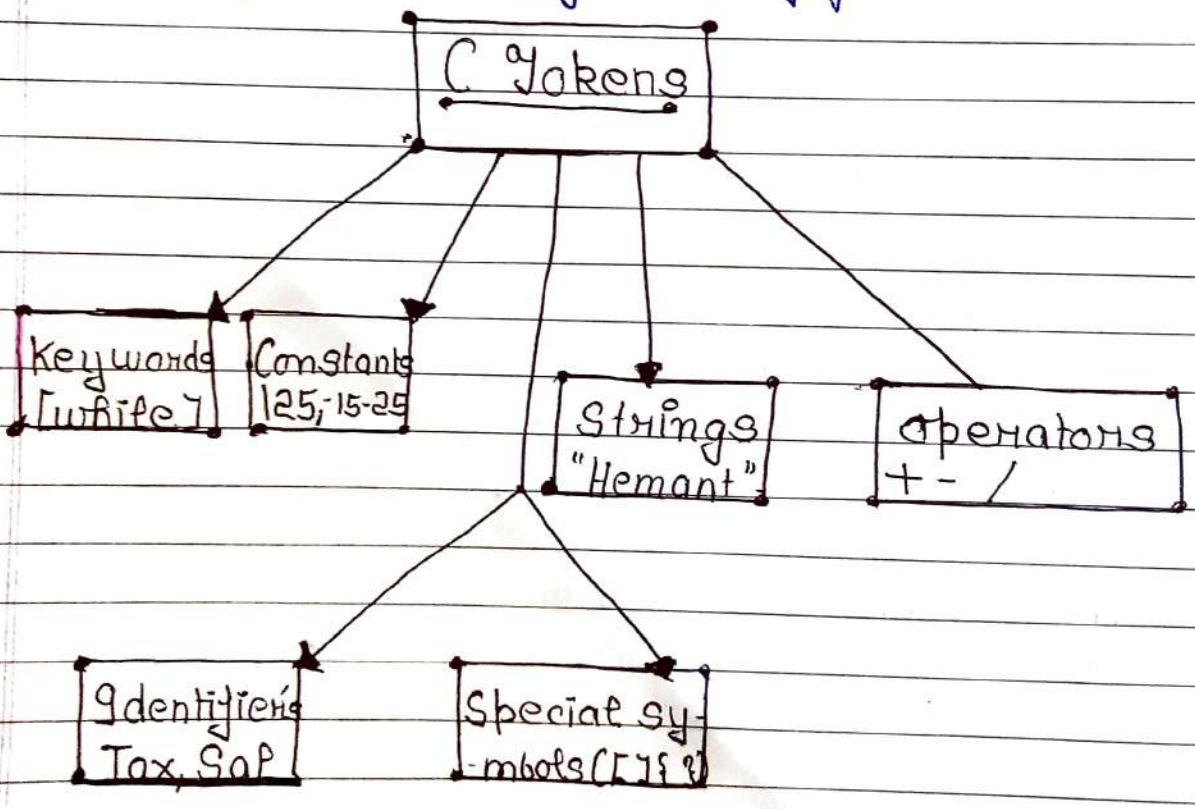
2.1 Digits :-

C character set supports digits from 0 to 9. Any combination of digits from this range is valid in C.

3.] Special characters :-

In addition to alphabets and digits. C support ~~so~~ many special chasacter's.

5.] C Tokens :- In C language, the smallest individual Units are called as tokens Tokens in C are of different types. These are represented by below figure.



[Types of C Tokens]

6.] C Keywords :- Every Word in a program of C language will be a keyword or an identifier. C compiler has 40 Keywords. The list of these keywords are as follows

C Keywords

auto	break	case	char	const	Continue	
default	do	double	else	enum	extern	
Float	For	For	goto	if	Int	
long	near	register	return	short	Signed	
static	struct	Switch	typedef	union	Unsigned	
void	while					

C has 32 general purpose and 8 speci-al purpose keywords.

5.1 Rules For declaring Identifiers :-

1. The first character of an Identifier must be an alphabet or underscore.
2. It can have letters, digits and valid special characters.
3. C will take only first 31 characters of the Identifier.
4. Identifier cannot be a keyword
5. You cannot use a white space in the Identifier

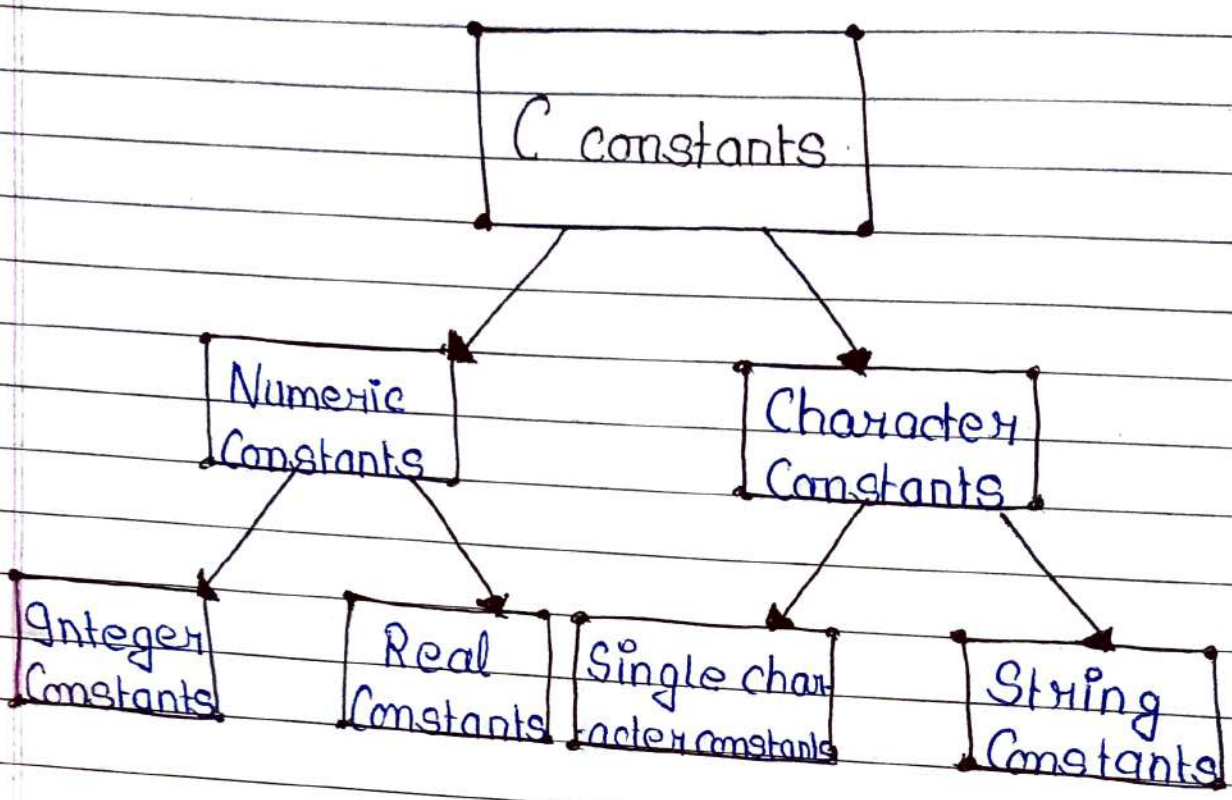
7.1 Constants :-

Constant refers to a value that do not change during the execution of a program. For Example mathematical term "Pie" is always assumed to be equal to 22/7 or 3.14.

eg $6x + 4y = 95$

In above example 6, 4, 95 values will never change

8.4 Types of C Constants:-



1.4 Numeric Constants:- These are the constants that are formed by combination of digits from 0 to 9. they may or may not have a decimal point in them.

i.4 Integer Constants:- Integer constants are the sequence digits that are used in C programs. Decimal integer, octal integers and hexadecimal integers are the three types of numeric constants i.e Octal integer constants consists of combinations of digit from 0 to 7. These are preceded by leading 0 or O. for example 025, 05245.

ii.4 Hexadecimal integer constants are represented by combinations of digits from 0 to 9 and

-racter s from A to F or alone - 0 to 15 Digit from 10 to 15 are represented by characters A to F respectively.

Rules For Constructing Integer constants :-

1. An Integer Constant must have atleast one digit
2. It must not have a decimal point.
3. It can be either positive or negative
4. It is assumed to be positive, if there is no sign specified.
5. Spaces and Commas are not allowed within it.
6. It can have values from -32,768 to 32768
   Example : 550, -665

2. **Real Constants** : Real constants are used to rep-resent the quantities that vary rapidly. The examples of such quantities are distances, weights, heights and prices. Real constants are also called as floating point Constants. Real constants are further of two types :-
   1. Real Constant in fractional form
   2. Real Constant in exponential form.

Rules for Construction of Real Constants in fractional form :-

1. It must have atleast one digit
2. It must have a decimal point
3. It could be either positive or negative.
4. Positive is considered as the default sign.

5.] Commas and Spaces are not allowed in real Consta
Example : 54-564, -65-235

Real Constants in fractional form are not adequate for representing very small and very large numbers. Due to this exponential form are used - for Example 3500000000 can be written as 3.5E9 when represented in exponential form a real constant contains two parts : mantissa and the exponent

Mantissa :- Part appearing before e is called as Mantissa

Exponent :- Part appearing after e is called as exponent.

for Example 125.3e5, 125.3 is mantissa part and 5 is exponent part.

Rules for Construction of Real Constants in exponential form :-

1.] The mantissa and exponent part must be separated by character e
2.] The mantissa may be positive or negative
3.] The exponent must Contain number which must be an Integer of any sign
4.] The default sign for both mantissa and the exponent is positive
5.] It can have value from range 3.4e38 to 3.4e

Some valid real constants expressed in exponential form are +2.8e-4, 2.3ee2, -0.5e-

2.] Character Constants :- In addition to numeric constants, the program also need to deal with characters and strings. Character Constants are used to express quantities like name, place or gender. They can be a single character or a group of characters or backslash character Constants.

i.] Single character Constant :- Any character or digit within single quotes will be treated as single character constant. Some valid character Constants are: 'a', 'c', '1', '2', 'X'

ii.] String character Constants :- A group of characters (constant contains a single charact.) is called as string. These constants are enclosed with double quotation marks. some valid string Constants are "Hemant", "shiva", "7898", "5+7". any group of characters or digits within double quotes will be treated as string

iii.] Backslash character constants :- In addition to above described character Constants, C Language supports some special character constants that are called "Escape sequences". Backslash character constants or

escape sequences are used with output function of C. Each of these characters has a specific purpose. They have been illustrated in following table

Constant	Meaning	Constant	Meaning
'\a'	Bell alert	'\v'	Vertical tab
'\b'	Backspace	'\''	Single quote
'\F'	Formfeed	'\"'	Double quote
'\n'	New line	'\?'	Question mark
'\r'	carriage Return	'\\'	Backslash
'\t'	horizontal Tab	'\0'	Null

9.7 Variables:- In C programming a variable is a container (storage area) to hold data. Variable is a name, given to a temporary memory location that has been used to store any value. (during program execution A variable is needed) It is clear from the name that value stored in variable can be changed at any instance of time during the program execution

example $5x+3y=77$

In above equation x and y are variables, which can have any values 5, 3, 77 are constant

Rules for Construction of Variables :-

The name of the variable can be any combination of single to 31 alphabets, digits or Underscore. These rules may vary with some

compilers

2. Variable name must start with a character.

3. Blank spaces and commas are not allowed within a variable name.

4. No special symbol, except the underscore (_) is allowed

5. The variable name cannot be same as any key-word.

6. Some examples of valid variable names are: H_1, hem, salary, tabill, bro etc

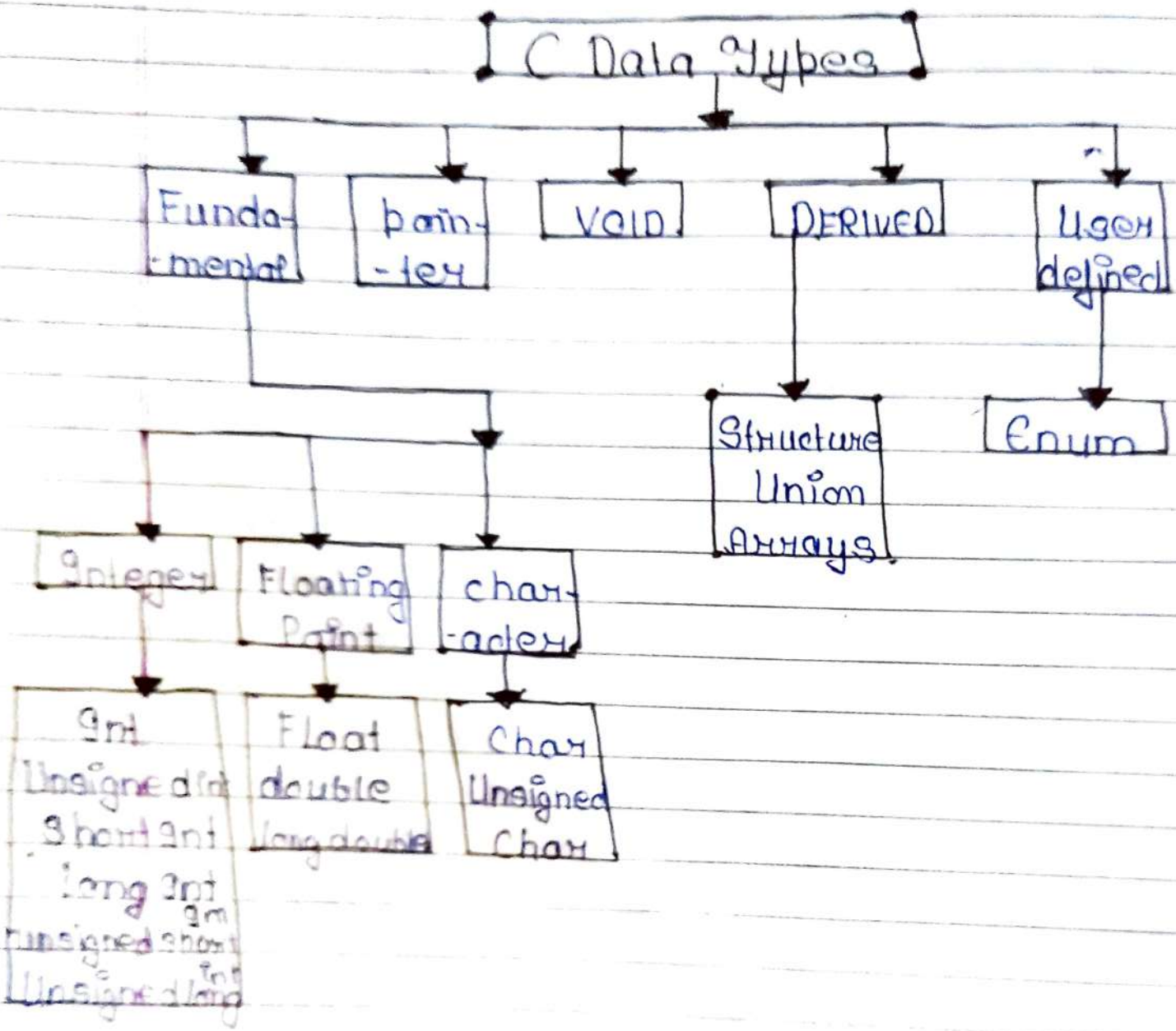
10. Variable Declaration :- Variable is declared by using a statement that contains the data type of variable and name of variable. This statement can also contain the initial value of variable. For example following statement declares a variable basic_sal of Integer type and initialises it to 3400

```
int basicsal = 3400;
```

11. Datatype :- Datatype represents type of input to be carried in the variable. The data types can be classified into primary or Fundamental data types, derived data types and User defined data types. Pointers and Void data types are other supported data types by C language

1. Primary data types :- Four fundamental data-types are supported by all C compilers. These are 1. Integer

2) Character 3) Float 4) Double precision float



1) Integer data type:- Integer data types are used to store numeric data items. They can store whole numbers as well as numbers with fractions. The data types of this category are Integer, short Integer, long Integer. In addition to these types unsigned int, unsigned short int and unsigned long int data types also fall under this category. Following table describes all Int data types in.

	Datatype	Description	Range	Bytes
1.	Int	Stores Integer (non-fractional) values both +ve and -ve	-32768 to +32,768	2
2.	Unsigned Int	Stores Integer (non fractional) values	0 to 65,535	2
3.	Short int	Stores Integer (non fractional) values both +ve and -ve	-128 to 127	1
4.	Unsigned Short Int	Stores Integer non fractional values	0 to 255	1
5.	Long int	Stores Integer values +ve and -ve	-2,147,483,648 to 2,147,483,647 (two billion one hundred forty seven million)	
6.	Unsigned long Int	Stores Integer values	0 to 4,294,967,295	4

2. Floating Point data type:- Floating Point data types are used to store numbers with fractions. following tables describes all floating point data types.

Datatype	Description	Range	bytes
Float	used to store fractional values	$3.4 * 10^{-38}$ to $3.4 * 10^{38}$	4

3. Double data types :- Float data type is not capable of storing very large numeric and floating quantities. for this double precision data type is used. following double data types described by below table.

DATATYPE	Description	Range	Bytes
double	Stores fractional values	$3.4 * 10^{-308}$ to $3.4 * 10^{308}$	8
long double	Stores fractional values	$3.4 * 10^{-4932}$ to $3.4 * 10^{4932}$	10

4. Character data types :- character data types are used to store character data items. The values to be stored in them are enclosed in single quotation marks (' '). char and Unsigned char are two character data types

Datatype	Description	Range	Bytes
char	Stores character values	-128 to 127	1
unsigned char	stores character values	0 to 255	1

2. Derived data types :- Derived data types are also called as structured data types. The main difference between fundamental and derived data types is that a variable of derived data type can handle more than one values at a time - whereas a variable of fundamental data types can handle only one value at time. The data types in this category are arrays, strings, structure and unions.

3. User defined data types :- These are the data types that are defined by User - They contain user defined specified values. Enum keyword is used to declare data types of this type.

4. Pointer data types :- Pointer is a data type that handles the data as per its memory address for example an Integer Variable holds an Integers value however Integer pointer holds address of Integer Vari

5. Void data type :- Void data type represents empty or null. This datatype is used when a particular function does not return any value.

Compiled and executed

7.] To see the output Press Alt + 5 from keyboard

6.] Errors in C Language:- Errors are the problem on the faults that occur in the program. There are mainly five types of errors exist in C Programming:

1.] Syntax Error
2.] Run time error
3.] Linker error
4.] Logical error
5.] Semantic error

1.] Syntax Error :- Errors that Occur when you vio-te the rules of writing C syntax are known as Syntax errors. All these errors are detected by Compiler and thus are known Compile time error's. Most Frequent Syntax er-ors are

1] Missing Parenthesis (})
2] Printing the value of variable without declaring it.
3] Missing Semicolon like this.

```
// C program to illustrate
// Syntax Error
#include <stdio.h>
void main()
{
int x = 10;
int y = 15;
Printf("%d", (x, y)
}
```

Error

Error : expected ';' before '}' token

Run time error :- Errors which occur during Pro-gram execution (run-time) after successful compilation are called run time error's - One of the most common run time error is division by zero also known as divison error. These types of error are hard to find as the compiler does not point to the line at which the error occurs. Example of run time error is :-

```
// C program to illustrate
// run time error
# include <stdio.h>
Void main ()
{
int n = 9, div = 0;
// Wrong Logic
// number is divided by 0
div = n/0;
Printf ("result = %d", div);
}
```

Error will appear as :-

Warning division by zero [-wdiv-by-zero]
div = n/0;

Linker Errors :- These error occurs when after compi-lation we link the different object files with main's object using ctrl + F9 key (RUN). These are errors generated when the

executable of the program cannot be generated. This may be due to wrong function prototyping, incorrect header files. one of the most common linker error is writing Main() instead of main(). Example of linker error is :-

```
// C program to illustrate
// linker error
# include <stdio.h>
Void Main() // Here Main() should be main()
{
int a = 10;
Printf("%d", a);
}
```

Error :-

(.text+0x20); undefined reference to 'main'

4. Logical error's :- On compilation and execution of program desired output is not obtained when certain input values are given. These types of errors which provide incorrect output but appears to be error free are called logical errors. These are one of the most common errors done by beginners of programming.

Example of logical error are :-

```
// C program to illustrate logical error
int main()
{
int i = 0;
// logical error : a semicolon after loop
```

```
For(i=0; i<3; i++);
{
printf("loop");
Continue;
}
getchar();
Return 0;
}
```

5.] Semantic Error's :- This error occurs when the statements written in the program are not meaningful to the compiler. Example of Semantic Error is

```
//C program to illustrate Semantic error
Void main()
{
int a,b,c;
a+b=c; //semantic error
}
```

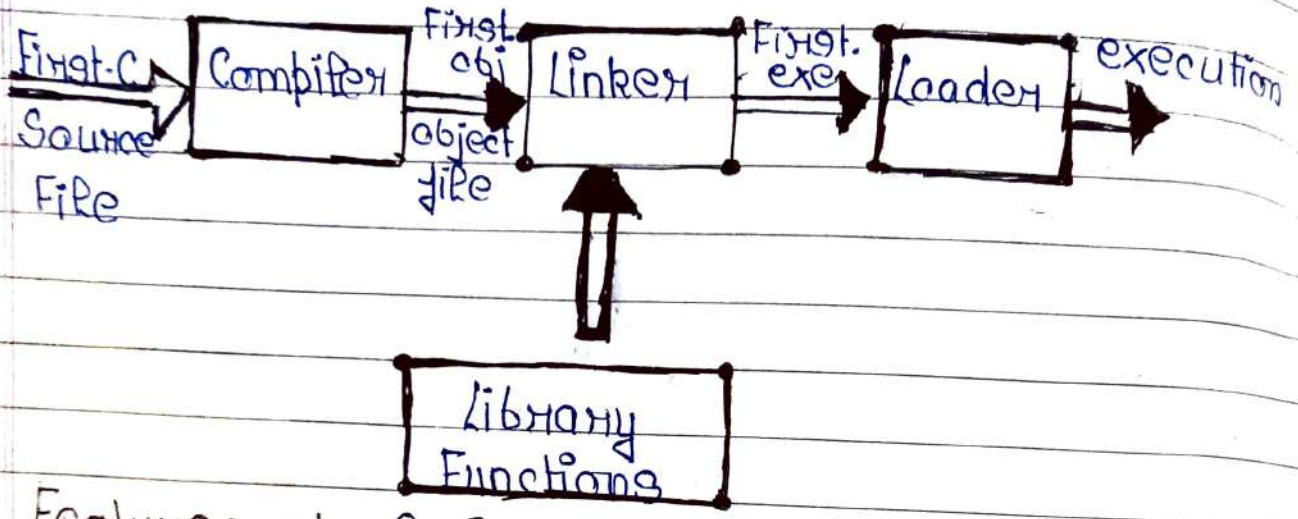
Error :-

error : lvalue required as left operand of assignm-ent a+b=c; // Semantic Error

17.] How does a program Executes :- whenever a C Program file is compiled and executed, the Compiler generates some files with same name as that of C Program file but with different extensions

P

Below figure shows the compilation process with the files created at each step of the Compilation process.



Features of C Programming :- The main key features of C Programming are as follows :-

1. Speed :- C comes with support for system programming and hence it compiles and executes with high speed when compared with other HLL languages.

2. Flexibility :- The second feature of C programming is flexibility. Flexibility means possibility of programmer to control the language

3. Modularity :- Possibility to break down large programs into small modules using sub-routine.

4. Portability :- It is a platform Independent programming language

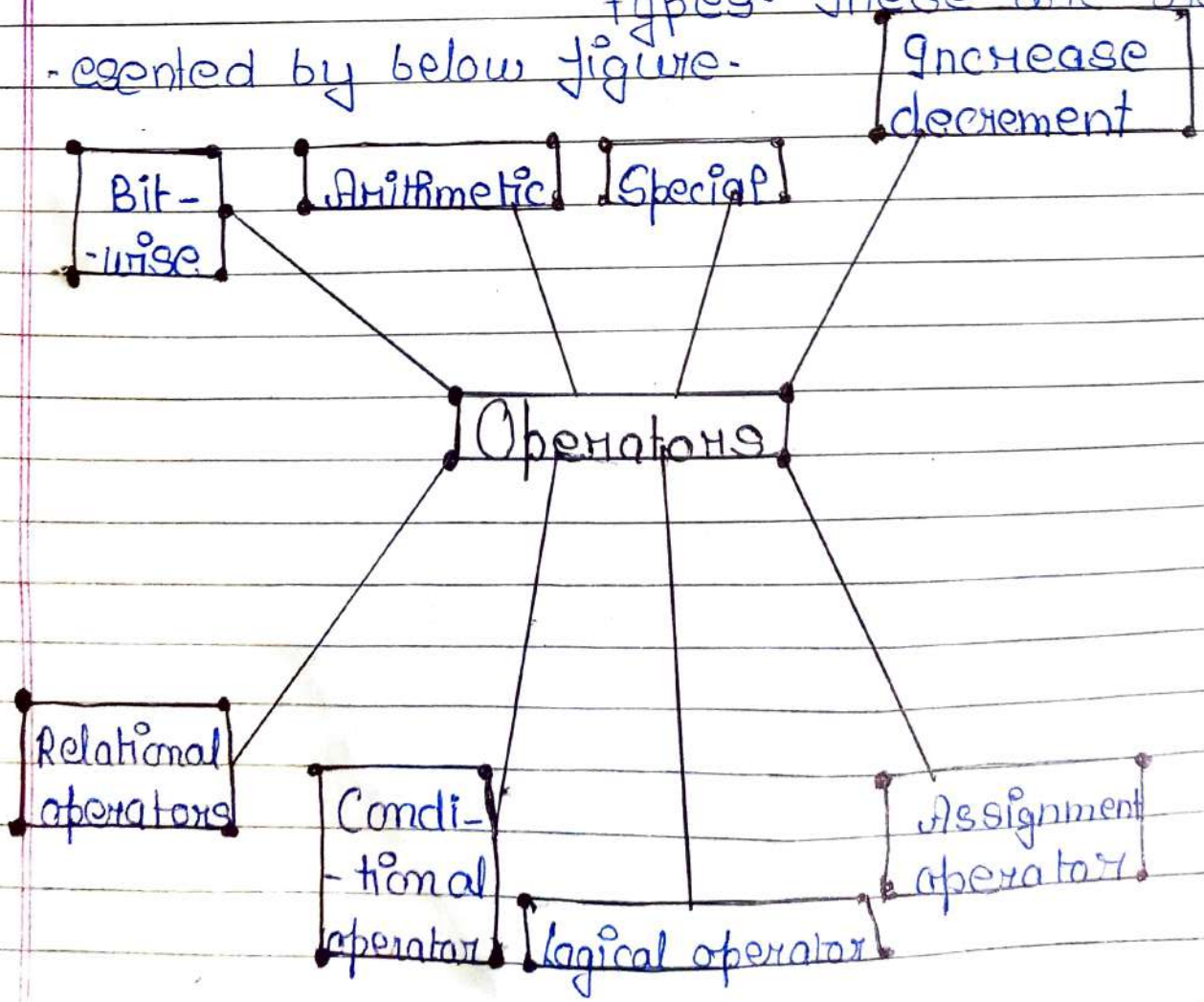
5. Extensibility :- Possibility to add new features by programmer.

Chapter - 3
Operators and Expressions.

1.] Operators :- An operator is a symbol that instructs the Computer to perform a specific operation. This operation varies with the type of operator. The main use of operators is to manipulate the data and variables.

2.] Expression :- The statement containing variables and operators is called as an expression.

3.] Types of operators :- Operators are of eight types. These are represented by below figure.



1.) Arithmetic operators :- C language supports arithmetic operations on variables through various arithmetic operators. following are the operators to perform arithmetic operations.

a.) Addition operator :- Addition operator "+" is used to add two or more integers or floats.

b.) Subtraction operator :- Subtraction operator "-" is used to calculate the difference of two numbers or floats.

c.) Multiplication operator :- Multiplication operator "*" is used to calculate the product of two or more integers or floats.

d.) Division operator :- Division operator "/" is used to calculate the quotient after dividing two number. This operator has no concern with remainder, left after division.

e.) Modulas operator :- The modulas division operator "%" is used to calculate the remainder left after the division of two numbers. This operator has no concern with quotient left after division

2.4 Relational operators :- Relational operators are used to compare two values Comparison is very important in a program and it acts as a base for executing certain statement in program. An expression containing one or more relational operators is called as relational expression. Relational operators are

i.4 Equal to operator (==) :- is used to compare two quantities for equality. The expression will return true if compared variables contain equal values. otherwise expression result FALSE.

2.4 less than operator (<) :- Compares the variables and results TRUE, if the value in the variable on left side of operator is less than value of variable present on the right hand side of the operator

3.4 less than or Equal to operator :- (<=) Compares the quantities and results true if value of variable on left side of operator is less than or equal to the value of variable present on the right hand side of the operator.

4.4 Greater than (>) :- Compares the variables and results true if value in a variable on left side of operator is greater than the value of variable present on the right hand side of operator.

5.4 Greater than or equal to operator :- (>=) Compares

the quantities and results true, if value in the variable on left side of operator is greater than or equal to the value of variable present on the right hand side of the operator.

6-) Not equal to (!=) Compares the quantities and results TRUE, if the quantity on left side of the operator is not equal to the quantity present on the right hand side of the operator.

3-) Logical operators :- C language is equipped with a support for logical operators to join multiple conditional expressions. These operators are used to combine two or more relational expressions to return the combined results of the relational expressions depending upon the operator used. Logical operators form logical expressions, which result either true or false. Logical operator are 1-) And 2-) OR 3-) Not. Logical operators are used to connect relational expressions.

1-) AND operator :- Combines two or more relational expressions and results true when all the relational expressions individually return true otherwise. It returns false. It is represented by '&&' symbol. For example the the expression "(x<4) && (x<z)" will result TRUE if (x<4) is true and (x<z) is also true else, it will return false.

2-) OR operator :- Combines two or more relational expressions and results true when

either one or more of relational expressions, individually return True. It returns false when all individual relational expressions result into false. It is represented by '||' Symbol

iii) NOT operator:- is used to reverse the result of relational expression. It just negates the obtained result. It is represented by exclamation symbol '!' for example expression "!(x>4)" will result false. If x is greater than 4 and will result true, if x is less than 4. So result is exactly opposite to that of (x<4)

4) Assignment operators:- "=" is called Simple Assignment operator of C language. It is used to assign values to variables in C. In C language, the variable on the left hand side of the assignment operator receives the value of variable or expression present on the right hand side of assignment operator. For example in the expression x=5 5 will be stored in variable "x" In expressions "a=z+t" or "a=c+10", result of expression will be stored in "a".

5) Increment and Decrement operators:- In addition to standard arithmetic operators, C also provides special operators for incrementing and decrementing the value stored in variable. ++ is called increment operator and -- is called decrem