**Unit 3 :Structure of C Language program**

1) Comment line

2) Pre-processor directive

3) Global variable declaration

4) main function()

{

Local variables;

Statements;

}

User defined function

}

}

**Comment line**

It indicates the purpose of the program. It is represented as

/……………………………../

**Comment line is used** for increasing the readability of the program. It is useful in explaining the program and generally used for documentation. It is enclosed within the decimetres . Comment line can be single or multiple line but should not be nested. It can be anywhere in the program except inside string constant & character constant.

**Global Declaration:**

This is the section where variable is declared globally so that it can be access byall the functions used in the program. And it is generally declared outside the

**function:**

main()

It is the user defined function and every function has one main() function from where actually program is started and it is encloses within the pair of curly braces.

The main() function can be anywhere in the program but in general practice it is placed in the first position.

**Syntax:**

main ()

{

……..

……..

……..

}

The main() function return value when it declared by data type as

int main()

{

return 0;

}

The main function does not return any value when void (means null/empty) as

void main(void ) or void main()

{

printf (“C language”);

}

Output: C language

The program execution start with opening braces and end with closing brace.

And in between the two braces declaration part as well as executable part is

mentioned. And at the end of each line, the semi-colon is given which indicates

statement termination.

/**First c program with return statement/**

#include <stdio.h>

int main (void)

{

printf ("welcome to c Programming language.\n");

return 0;

}

**Output:** welcome to c programming language.

**Steps for Compiling and executing the Programs**

A compiler is a software program that analyzes a program developed in a particular computer language and then translates it into a form that is suitable for execution.

**character set**

A character denotes any alphabet, digit or special symbol used to represent information. Valid alphabets, numbers and special symbols allowed in C areThe alphabets, numbers and special symbols when properly combined form constants, variables and keywords.

**Identifiers**

Identifiers are user defined word used to name of entities like variables, arrays, functions, structures etc. Rules for naming identifiers are:

1) name should only consists of alphabets (both upper and lower case), digits

and underscore (\_) sign.

2) first characters should be alphabet or underscore

3) name should not be a keyword

4) since C is a case sensitive, the upper case and lower case considered

differently, for example code, Code, CODE etc. are different identifiers.

5) identifiers are generally given in some meaningful name such as value,

net\_salary, age, data etc. An identifier name may be long, some implementation

recognizes only first eight characters, most recognize 31 characters. ANSI

standard compiler recognize 31 characters. Some invalid identifiers are 5cb, int,

res#, avg no etc.

**Keyword**

There are certain words reserved for doing specific task, these words are known as reserved word or keywords. These words are predefined and always written in lower case or small letter. These keywords can’t be used as a variable name as it assigned with fixed meaning. Some examples are int, short, signed, unsigned, default, volatile, float, long, double, break, continue, typedef, static, do, for, union, return, while, do, extern, register, enum, case, goto, struct, char, auto, const etc.

**Data types**

Data types refer to an extensive system used for declaring variables or functions of different types before its use. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted. The value of a variable can be changed any time.

**C has the following 4 types of data types**

**basic built-in data types:** int, float, double, char

**Enumeration data type:** enum

**Derived data type:** pointer, array, structure, union

**Void data type:** void

* A variable declared to be of type int can be used to contain integral values only—that is, values that do not contain decimal places.
* A variable declared to be of type float can be used for storing floating- point numbers (values containing decimal places).
* The double type is the same as type float, only with roughly twice the precision.
* The char data type can be used to store a single character, such as the letter a, the digit character 6, or a semicolon similarly A variable declared char canonly store character type value.

***There are two types of type qualifier in c***

**Size qualifier**: short, long

**Sign qualifier:** signed, unsigned

**Constants**

* Constant is a any value that cannot be changed during program execution.
* In C, any number, single character, or character string is known as a constant.
* A constant is an entity that doesn’t change whereas a variable is an entity that may change.
* For example, the number 50 represents a constant integer value. The character string "Programming in C is fun.\n" is an example of a constant character string.

**C constants can be divided into two major categories:**

1. Primary Constants
2. Secondary Constants

**Variables**

* Variable is a data name which is used to store some data value or symbolic names for storing program computations and results.
* The value of the variable can be change during the execution.
* The rule for naming the variables is same as the naming identifier.
* Before used in the program it must be declared. Declaration of variables specify its name, data types and range of the value that variables can store depends upon its data types.

**Syntax:**

int a;

char c;

float f;

**Variable initialization**

* When we assign any initial value to variable during the declaration, is called initialization of variables.
* When variable is declared but contain undefined value then it is called garbage value.
* The variable is initialized with the assignment operator such as

**Data type variable name=constant;**

**Expressions**

An expression is a combination of variables, constants, operators and function call.

It can be arithmetic, logical and relational

for example:-

int z= x+y // arithmatic expression

a>b //relational

a==b // logical

func(a, b) // function call

Expressions consisting entirely of constant values are called constant expressions. So, the expression

121 + 17 - 110

is a constant expression because each of the terms of the expression is a constant value. But if i were declared to be an integer variable, the expression

180 + 2 – j

would not represent a constant expression.

**Operator**

This is a symbol use to perform some operation on variables, operands or with the constant. Some operator required 2 operand to perform operation or Some required single operation .Several operators are there those are, arithmetic operator, assignment, increment ,decrement, logical, conditional, comma, size of , bitwise and others.

**1. Arithmatic Operator**

This operator used for numeric calculation.

**2.Assignment Operator**

A value can be stored in a variable with the use of assignment operator. The assignment operator(=) is used in assignment statement and assignment expression.

**3.Increment and Decrement**

The Unary operator ++, --, is used as increment and decrement which acts upon single operand. Increment operator increases the value of variable by one .Similarly decrement operator decrease the value of the variable by one

**4.Relational Operator**

It is use to compared value of two expressions depending on their relation. Expression that contain relational operator is called relational expression.

**5. Sizeof Operator**

Size of operator is a Unary operator, which gives size of operand in terms of byte that occupied in the memory. An operand may be variable, constant or data type qualifier.

**6. Bitwise Operator**

Bitwise operator permit programmer to access and manipulate of data at bit level.

Various bitwise operator enlisted are

* one's complement (~)
* bitwise AND (&)
* bitwise OR (|)
* bitwise XOR (^)
* left shift (<<)
* right shift (>>)

These operator can operate on integer and character value but not on float and double. In bitwise operator the function show bits( ) function is used to display the binary representation of any integer or character value.

**Logical or Boolean Operator**

* Operator used with one or more operand and return either value zero (for false) or one (for true).
* The operand may be constant, variables or expressions. And the expression that combines two or more expressions is termed as logical expression.
* C has three logical operators :

Operator Meaning

* && AND
* || OR
* ! NOT

Where logical NOT is a unary operator and other two are binary operator.

Logical AND gives result true if both the conditions are true, otherwise result is false.

And logical OR gives result false if both the condition false, otherwise result is true.