C PROGRAMMING

a computer program is just a collection of the instructions necessary to solve a specific problem.

The basic operations of a computer system form what is known as the computer’s instruction set.

And the approach or method that is used to solve the problem is known as an algorithm.

So for as programming language concern these are of two types.

1) Low level language

2) High level language

Low level language:

Low level languages are machine level and assembly level language.

In machine level language computer only understand digital numbers i.e. in the form of 0 and 1.

So, instruction given to the computer is in the form binary digit, which is difficult to implement instruction in binary code.

This type of program is not portable, difficult to maintain and also error prone.

The assembly language is on other hand modified version of machine level language. Where instructions are given in English like word as ADD, SUM, MOV etc.

It is easy to write and understand but not understand by the machine.

So the translator used here is assembler to translate into machine level. Although language is bit easier, programmer has to know low level details related to low level language.

In the assembly level language the data are stored in the computer register, which varies for different computer.

Hence it is not portable.

High level language: These languages are machine independent, means it is portable.

The language in this category is Pascal, Cobol, Fortran etc.

High level languages are understood by the machine.

So it need to translate by the translator into machine level.

A translator is software which is used to translate high level language as well as low level language in to machine level language.

Three types of translator are there:

Compiler

Interpreter

Assembler

Compiler and interpreter are used to convert the high level language into machine level language.

The program written in high level language is known as source program and the corresponding machine level language program is called as object program.

Both compiler and interpreter perform the same task but there working is different.

Compiler read the program at-a-time and searches the error and lists them.

If the program is error free then it is converted into object program.

When program size is large then compiler is preferred.

Whereas interpreter read only one line of the source code and convert it to object code.

If it check error, statement by statement and hence of take more time.

Integrated Development Environments (IDE)

The process of editing, compiling, running, and debugging programs is often managed by a single integrated application known as an Integrated Development Environment, or IDE for short.

An IDE is a windows-based program that allows us to easily manage large software programs, edit files in windows, and compile, link, run, and debug programs.

On Mac OS X, CodeWarrior and Xcode are two IDEs that are used by many programmers.

Under Windows, Microsoft Visual Studio is a good example of a popular IDE.

Kylix is a popular IDE for developing applications under Linux. Most

IDEs also support program development in several different programming languages in addition to C, such as C# and C++.

Structure of C Language program

1 ) Comment line

2) Preprocessor 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 decimeters.

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.

Preprocessor Directive:

#include tells the compiler to include information about the standard input/output library.

It is also used in symbolic constant such as #define PI 3.14(value).

The stdio.h (standard input output header file) contains definition &declaration of system defined function such as

printf( ), scanf( ), pow( ) etc.

Generally printf() function used to display and

scanf() function used to read value Global

Declaration: This is the section where variable are declared globally so that it can be access by all 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 int main (void)

{ printf ("welcome to c Programming language.\n"); return 0; }

Output: welcome to c programming language.

Character set

A character denotes any alphabet, digit or special symbol used to represent information. Valid alphabets, numbers and special symbols allowed in C

The alphabets, numbers and special symbols when properly combined form constants, variables and keywords.