C Language:

* Dennis Ritchie is the father of C Language

Flavours of C: Platform:

* ANSI C unix
* Dynamics C linux
* Turbo C MS Doc
* Borland C Windows
* Microsoft C Windows
* Pro C Oracle
* Objective C Mac

Steps:

1. Algorithm
2. Flow chart
3. Program

Program : -

* set of instructions to reach a particular task

Sample program :

#include<stdio.h> // pre processor directive(1st line of code)

#include<math.h>

#include<string.h>

#include<graphics.h>

#include<stdlib.h>

#include<alloc.h>

#include<ctype.h>

main()

{

printf(“hello world”);

}

* Main()

Note: What is the entry point of C program?

Ans: main is the entry point of C program

* When a program is created it is stored as

1. Name.c -source code
2. Name.c -backup code
3. Name.obj -object file(intermediate conversion into machine language)
4. Name.exe -executable file

* Compiling a file

linux/unix:

First .c

Cc first.c → compiling

./a out → running

* Default return typr of function in c is int
* I.e main() = int main()
* Void main() returns nothing/null
* Int main()
* Or main() return 0// return(0);
* Int main() is preffered to void main()
* Data types:

-----------------------

Int %d

Char %c (or) %s

Float %f

long int %ld

Double %If

-char can take integers

- i.e char num[3] → 001 , int only displays

- **Typecasting** is process of converting one data type to another

1)implicit type casting

2)explicit typecasting

1. Implicit typecasting example:

#include<stdio.h>

main()

{

int val=65;

char ch='a';

printf("\n val =%c",val);

printf("\n ch=%d",ch);

}

1. Explicit typecasting examples:

#include<stdio.h>

main()

{

int m,p,c,tot;

float avg;

printf("enter the marks of maths,physics,chemistry");

scanf("%d%d%d",&m,&p,&c);

tot=m+p+c;

avg=tot/3;

printf("\n total marks: %d",tot);

printf("\n average: %f",avg);

}

→ Control statements:

* If
* If else
* Else if

→ Case control statements

* switch
* If :- used to check a condition and executes the statements if it is true.

→ Syn :-

if(condition)

{

Statements……

}

* If else :-

→ syn :-

it(condition)

{

Statements….

}

else(condition)

{

Statements…

}

Ex: given number is even or odd

#include<stdio.h>

int main()

{

int n;

printf("\n enter a number:");

scanf("%d",&n);

if(n%2==0)

{

printf("even number");

}

else

{

printf("%d is odd ,n");

}

return 0;

}

* Else if:

→ syn:

if(condition1)

{

Statement

}

elseif(condo2)

{

Statement;

}

Else

{

Statement;

}

Ternary operators ? :

Condition? Statement if true : statement if odd

Eg: printf(n%2==0? ”Even” : “odd”)

printf(n<0?”-ve”:n==0? “zero”:”positive”)

→ Operators in C :-

--------------------

* Arithmetical

+

-

\*

/(division)

%(modulus….remainder)

* Relational:

<

>

<=

>=

==

!=

Swaping of two numbers:

#include<stdio.h>

void main()

{

int a,b,c;

printf("enter two values a and b");

scanf("%d%d",&a,&b);

c=a;

a=b;

b=c;

printf("%d%d",a,b);

}

* Escape sequence:
* \n - new line
* \t - new tab
* \a - alert
* \b - backspace
* \\ - \
* \\* - double quotes
* \’ - single quotes

Example:

#include<stdio.h>

int main()

{

printf("\"hello world\"")

}

o/p:”hello world”(with semi colon)

Ex: even or not

#include<stdio.h>

int main()

{

int n;

printf("\n enter a number:");

scanf("%d",&n);

if(n%2==0)

{

printf("even number");

}

return 0;

}