**C**

Documentation: comprise of nameof program, developer. Compiler will not execute. Starts with \*/ and ends with \*/

Link: C have many libraries. Eg., stdio

Definition: constants can be defined.

Global declaration: variable that is declared outside the main function. Accessible throughout the program and all functions

Main() Function: starts execution from this

{

Declaration part: declare a variable that is accessible only in main method

Executable part: every statement must end with 0.

}

Subprogram: user defined functions

Function1

Function2

..

Ex:

/\* Simple program in C”\*/

#include<stdio.h>

Int main()

{

printf(“\n Welcome to C”);

return 0;

}

Flowchart:

Circle/oval=start/stop

Parallelogram=input/output

Diamond: decision

Rectangle=statement

Compile and Execute:

1. Create the program
2. Compile the program into human understandable to machine language
3. Link with C library
4. Run the program

* Program name as sample.c
* GNU C compiler: gcc sample.c
* Execution by defaut: a.out

Character ser: 256 characters

* Letters: all uppercasr and lowercase letters
* Digits: all decimal digits 0 to 9
* Special characters: To perform mathematical operations, condition checking, white/black spaces

+-\*/.<>:”’?()[]{}&%!@^$#

C Tokens:

* Every smallest individual unit. Eg., printf
* Every instruction is collection of tokens
* To construct c program.
* It can be keyword, operator, special symbls, strings, constants.

C keywords:

* Specual symbols with pre defined meaning
* Building blocks
* 37 keywords
* All in lowercase. It Is case sensitive

1. Auto: automatic storage clss
2. Break: is an unconditional control stataemnt to stop the loop.
3. Case: swith case
4. Char: character datatype
5. Const: constant value or barivale
6. Continue: is an unconditional control statement to go to the beginning f loop
7. Default: sewitch case statement to give a
8. Do: block in do while
9. Double: data type
10. Else: if else, else is false
11. Enum: enumerator data type.
12. Extern: external storage class
13. Float: datatype floating point
14. For: looping statemt
15. Goto: unconditional control statemtn.
16. If:
17. Long
18. Int
19. Register: storage class
20. Return: return any value
21. Short:
22. Signed: datatype
23. Sizeof: operatot to get the size
24. Static: storage class to provide constant value
25. Struct: create a rogeam
26. Switch: switch case
27. Typedef: user defined datatype
28. Union: group unsimilar elemetns
29. Unsigned:
30. Void: doest return anything
31. Volatile: usre to create
32. While: looping statement

Identifiers:

Refers to the name of variables/funitons/pointers/ array or structures

Ex: int ***marks***;

Char ***studnetName***[30]

Rulrs for Identifies:

* First characer must be an alphabet ot \_
* Should not start with number
* Should not use special characters between identifiers
* Only 31 characters weill be considered
* Keyword should not be used
* Contain only alpha num and \_

Constants:

Fixed value

Donot change

Type of constants: Numeric and Characters

Numeric: integer and real

Character: single and string (represented by ASCII)

Integer: sequence of digits

Can be decimal/octal/hexadecimal

* Decimal: 0 to 9

Ex: 123,-456,0,97879,+89

* Octal: 0 to 7 with leading o or O

Ex: o73,

* Hexadecimal: preceded by ox or OX. 0 to 9 and A to F

Ex: OX5, Oxabc

Real: floating point constant must contain both intefer and decimal

Ex: 3.14, -9.44

Can be expressed in exponential:

Ex: 0.45e5

Single Character constant: enclosed in ‘ ‘

Ex: ‘A’

Character constant has max of one character

Backslash characters: for output fiuncutins known as escape sequences.

‘\n’: next line

‘\O’: null

String: collection of characters, digits, specual characters and escape sequence enclosed in “ “

We can have multiple lines

Ex: CHARAN\

ROXTA”

We can also define string constant by separating in with white space

Ex: “CHARAN” “ROXTA”

Creating constants in C

* Using ‘const’ keyword.

Syntax: const datatype constantName;

Ex: const int x=10;

* Using ‘#define’ preprocessor:

Syntax: #define CONSTANTNAME value

Ex: #define PI 3.14

Variables: name to memory location to store different values to same datatype

Storage container t hold values of same datatype

Name Should not start with numbrs

Should not be keywords

Can contain only alpha num and \_

Considers only first 31 charactets

Datatypes: set of vzlue with predefined charactrtistd

To declare varibalws, const,arrays, etxc

* Primary
* Dervid
* Enum
* Void(returns/stores nothing)

Primary:   
Integer: signed int(- and +)- int, short int, long int. and unsigned int(+)- int, short int, ;ong int

Floating point: float, double, long double

Character: char, signed char and unsigned char

Integer: whole numbers

Int to represrent

Ex.

#include<stdio.h>

Void main()

{  
 int x=10;

Printf(“x=%d”,x); /\*%d for integer datatypes\*/

}

Floating Point: decimal

Float to represrnt float

Doubtle to reresetn double

Difference in precision

Ex.

#include<stdio.h>

void main()

{  
 float x=10.434;

Printf(“x=%f”,x); /\*%f for floating point datatypes\*/

}

Character type:

Char- 8 bits

#include<stdio.h>

Void main()

{  
 char x=’t’;

Printf(“x=%c”,x); /\* %c for char\*/

}

Void types: no vales  
to specify type of functoins

Returns nothing

Enumerated Types: enum

Syntasx: enum identifier{value1, value2,….);

Ex:

enum day{Mon, Tue,…..Sun};

Derived(user defined)

* Arrays
* Structures
* Unions
* Functions
* Pointers

Declaration:

* Datatype variable\_name;

Ex: int number;

* Datatype variable1, variable2;

Ex: float sum, avg;

User defined: allows user to define teir own identifiers for existinf datatypes.

Ex: typedef int marks;

Marks maths[20], science[20]

Deckaration of storage class:

Provides info about location, scope, visibility.

Four: auto, extern, static(constants) and register(mostly used)

Auto: local variable known only to the function.

#include<stdio.h>

main(){

int x=10;

{  
 int x=20;

Printf(“x=%d”,x);

}

Printf(“x=%d”,x);

}

Extern: global

#include<stdio.h>

Int x=10;

Int main()

{

Printf(“x=%d”,x);

Return 0;

}

Static: local and retains its value. Default value is 0.

Initialized only once. Modified any no. of times

#include<stdio.h>

main()

{  
 static int x=10;

Printf(“%d”,x--);

If(var>0)

Main();

}

Register: to store local varibaless in cpu registers for quick access

#include<stdio.h>

main()

{  
 register int a,b;

scanf(“%d%d”,&a,&b);

printf(“%d %d”,a,b);

}

e