C PROGRAMMING :

DEVELOPMENT :

Types of applications :

Mobile applications : .apk ,need to install in Mobile phones

Websites : Run by URL(Uniform Resource Locator) : It runs on web-browser

Method to create -> Designing Language : HTML + CSS , BOOTSTRAP

Web Applications : Dynamic Websites : Runs by URL , It runs on web browser

Method to create :

- Designing : HTML +CSS / BOOTSTRAP
- Programming : C , C++, JAVA, C#, PYTHON , PHP, Javascript……………….
- Database : MYSQL, MSSQL, MSACCESS, ORACLE, MONGODB, ……………….

Console Applications : Run by Commands on blue/black screens

Desktop Applications : Run by .exe files , need to install

Standalone applications : Key based softwares , like Antiviruses

Languages : Languages is a medium to communicate with other person.

Programming Language : It is a medium to communicate with computer system.

By coding , you are able to perform any task by computer system.

C programming : Standalone Applications : to apply logics

C is a high level programming language. It was developed by Dennis Ritchie in 1972 in A&T's Bell Laboratory. C is a HLL(High level language) that means maximum words used in C programming is taken with the real life words.

High level language that means a developer friendly language.

Computer->Language-> Machine language(Binary Language /Low level language/0&1)

C is also known as mother language just because syntaxes of maximum other programming language is taken from the C.

Basic Structure of C programs :

1. **Documentation Section : (Optional)**
   Comments : Comments are those line of program , that is ignored by Compiler. It is just for the understanding of developer.
   Single Line Comment : //Comments…………….
   Multi-line comment : /* ………………………………….

………………………………………………………………*/

2. **Linking Section : Each programming language provides some pre-defined codes, that is saved in the library of programming.**
   When we need to use those pre-defined codes firstly we need to add library in the programs. : (Mandatory)
   C – Header Files
   Java – Package
   Python – Modules
   C# - Namespace
   Header File : Header file is the collection of some pre-defined functions. Functions are kept in different header files based on the category.
   stdio.h : Standard Input Output : printf(), scanf(), …………………….
   conio.h : Console Input Output : clrscr(), getch(), getche(), getchar(), putchar()…………………….
   math.h : pow(), sqrt(), log(), ceil(), floor()…………………
   String.h : strcpy(), strlwr(), strupr(), strrev(), strcat(), ……………………….
   #include"header_file_name_with_extension"
   #include< header_file_name_with_extension>
   # - Pre-processor directive
   Lines written with # is executed first in the program.
   include : include is a folder that contains all header files
3. **Global Declaration Section : Global declaration that means declaring some variables globally that is accessible in all function of programs : (Optional)**
   int a=5; //globally declared

```
    void main() //pre-defined function + user-defined function
    {
    A=a+10;

}

    Void techpile() //user-defined function
    {
    }
```

4. **Main Function : Main function is a mandatory function of each C programs.** Main function is declared with named main(). : (Mandatory)

```
void main()
{


}
//void – Data type
Main() - Function
```

5. **Other sub programs : You can define other user-defined function in C programs . It is optional. : (Optional)**

```
//This is my First Program //comment lines
#include<stdio.h> // Header-files
void main()   //it is the main part of program , execution starts from here
{

}
```

Keywords : Keywords are some reserve words that has special meaning and functionality . In C program total 32 keywords are defined . In programming keywords are used at special places whenever it needs.

Ex : if, else, do, int , while, float , double, long, struct, break, continue……..

Identifier : Identifiers are used for naming like : user defined function, variable, constant.

Rules to declare identifier :

Identifier can not start with Number. It can not be NumericAlpha.

Ex : 2var : invalid
Identifier can not have space in name .
It can not contain some special symbols like & , * etc
It can not be same as keywords
It can be Alpha-numeric like : var1, number1

Identifier : variables, constant , UDfunction
Data_type variable_name;
Data_type constantname;
Data_type function_name();

Data Types : Data types are some reserve words that defines the type of value and total memory space. Data type defines which type of value , it can store and how much memory space it will occupy.

There are mainly Four type of data type in C :

1. **Primitive data type / Basic / Fundamental Data Type**

Data type name	Format Specifier	Memory Space	Ex Values
Short : whole num	%d	2 byte	-32768 to + 32767
int : whole num	%d	2 byte/ 4 byte	3
long int : whole num	%ld	4 byte	
long long int : whole num	%Ld	8 byte	
Float : decimal	%f	4 byte	3.5
Double : decimal	%lf	8 byte	
long double : decimal	%Lf	10 byte	
char	%c	1 byte	

2. **Non-primitive data type / Derived data type : Derived data type** are created with the help of basic data types.
   Ex : String , Array , Pointer
3. **User-defined data type : User defined data type are declared by user. Which type of value it can store is decides by user.**

Ex : Structure , Union

4. **Empty data type: void is the empty data type. It is used to declare functions. It is not used to declare variables.**

Input & Output function in C : Each programming language some pre-defined input output function.

I/P Function : Input function is used to take input from the user at run time. Where you can store those values in a variable and can use anywhere in the program.

- Formatted Input Function :
  scanf() :
  scanf("format_specifier",variable_list_with_&); //int, float, long int , double, char , one word string "riya"
  long int v;
  scanf("%ld",&v);

- Un-formatted Input function :
  getch() : used to input a single character. getch() allows user to input only a single character. Without showing the inputed character , program will execute to the next line .
  This is unformatted function , specially made for the character input.
  Syntax :
  My_variable=getch();
  ----
  getche() : used to input a single character from user
  getchar() : used to input a single character from user
  gets() : used to input a multi-word string value from user

Output Function : O/p functions are used to print message or value on the screen for user.

- **Formatted Output function**

printf() :

```
void main()
{
printf("my variable is a");
}
```

- **Un-formatted output function**

putchar()

puts()

--

Basic Introduction of C

Basic structure of C program

Data type, keywords, variables, etc

--

Variable : Variables are some temporary storage area, which can hold a value and it's value may be change any where during the execution of program.

Note :

1. **Declaration of variable : Each variables in C, should be declare once at the top of program. Without declaration you can not use any variable in your program.**
   Syntax : data_type variable_name;
2. **Definition of variable /** assign value to the variable

a. Direct Initialization / compile time initialization
var=50;

b. User Input / Run time initialization
scanf("format_specifiers",variable_list_with_&);
scanf("%f",&var); - & - denotes the memory address of variable
int a; float f;
scanf("%d",&a);

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

c. **By expression / calculation**
A=b+10;

How many Variable =?
Variable declare
Variable initialize – value
Int a;
A=30; //
Scanf("%d",&a); //

User input :
Numeric – scanf()
Char – getch()

Exercises & Task:
Coding - error

Numeric value : whole number / real numbers – scanf

Character : getch(), getche(), getchar()

String : gets()

O/P FUNCTION :

Numeric value : whole number / real numbers – PRINTF()

SINGLE CHARCATER : PUTCHAR()

Character input special functions :

getch() : getch() is used to input a single character from user. Getch() function reads the inputed value from the screen and saves the value in a character type variable.

Working : getch() permits user to input only a single key on the output screen, next line of program executes without waiting to press enter key , as soon as user inputs a single character.

Syntax :

```
char ch;

ch=getch();
```

Character inputed by user, by using getch() function does not appear on output screen .

----

getche(): getche() also permits user to input only a single character and character inputed by user appears on the output screen and next line of program executes without waiting for enter key.

ex : //wap to input a character and print the inputed character

```
#include<stdio.h>

void main()

{

	char ch;

	printf("Please enter a single character : ");

	ch=getche(); //?
printf("\nInputed character is : %c",ch);

}
```

----

getchar() : getchar() function works same as the scanf() , but getchar() is a unformatted input function so you do not need to add any format specifier here.

Syntax :

```
char val;

val=getchar();
```

C source code -> Compile -> Object file -> converted into .exe file -> output

----

Operator : Operators are some special symbols , pre-defined in library , that is used for special functionality.
Each operator has it’s own use, when ever we need this we can use the operators in any statement of program.
Operators are always used with operands .
Suppose a statement :
A=A+b : here a,b is the operand and + is the operator
= is also a operator.
Unary operator : used with one operand , like Increment – decrement operator
Binary operator : used with minimum 2 operands like : +,-, *, > , < etc
Ternary Operator : used with minimum 3 operands like : conditional operator

Based on the working operators are divided into many category :

1. Arithmetic operator
2. Relational Operator
3. Logical operator
4. Assignment operator
5. Increment & Decrement operator
6. Conditional Operator
7. Bitwise operators

1. Arithmetic Operator : This operator is used for mathematical operations. It is used with numeric values. It is a binary operator that means it needs min. 2 operands to be used.

   + : addition

   - : subtract

   * : multiply

   / : division

   % : modular division

   Precedence of operator : (Which one will execute first)

   *

   /

   %

   +

   -

   =

   Example :

```
#include<stdio.h>
void main()
{
        int n1,n2,n3;
        printf("enter three numbers : ");

        scanf("%d%d%d",&n1,&n2,&n3);

        printf("sum of numbers is : %d",(n1+n2+n3));

        printf("\nsubtract of first & second number is : %d",(n1-n2));

        printf("multipky result of last two values is : %d",n3*(n1-n2));

    }
```

/ : it returns always a integer number if both operands are integer.

It returns a float value if any one operand is float.

/ : 20/3 = 6

20.0 /3 = 6.6666

% : It is called as modulo operator. It returns remaining value after the division of two integer type values. It can not be applied on floating type value.

Int n1=40, n2=9;

N1/n2=4

N1%n2=division remainder

Precedence of arithmetic operatr:

*,/,%: same precedence :high:Associativity :left to right

+,-: same precedence :Low:Associativity :left to right

Int a=a-10*b

Realational operators:relational operators are used to specify the relation between two operands.it is a binary operator.

Return type /result of relational statement is always true/false

Relational statement is a Boolean statement.

==:equal to

>:greater than

<:less than

>=:greater than equal to

<=less than equal o

!=:not equal to

Example.

A=a>10;

Conditional Operators: Conditional Operators are used to execute ane statement out of two statement based on a condition.

It is optional of if -else statement. When there is only line statement in if and else then you can use conditional statement in more easy way.

It is ternary operators, that means it executes with 3 operands.

? and :is condtional or ternary operators

Syntax-

Condition? true_statement : false _statement

Condition is a statement which result is always true or false

Logical Operator:-Logical operator is also used in two operands .return value of logical operator is true or false.

&& : Logical AND

|| : Logical OR

! : Logical not / Logical Not

!= : ignores a single value , a!=20,ch!='m'

== : stands for a single value a==20,char ch;ch=='z'

Statement	Output	Statement	Output
True && True	True/1	True||True	True/1
False && False	False/0	True|| False	True/1
True && False	False/0	False||True	True/1
False && False	False/0	False||False	False/0

1&&1

20 && 50

20>40 && 50>60

Assignment Operator- = is the assignment operator .it is the used to assign right hand side calculation result to the left hand side variable.

=comes with the lower precedence so it is done after the execution of statement.

A=a>b;a=?,a=0;

a+b=20; //invalid statement

in left hand side there should be a single variable