# Introduction to ‘C’ language

**1 Explain basic structure of ‘C’ program.**

| **Basic Structure of C Program is as** | **Example of C Program is as below,** |
|---|---|
| Documentation Section:<br><br>this section is optional, it include all documented related details of your program such as author name, program purpose , date of creation. This section is written in comment line | // This program is to find area of circle |
| Link Section :<br>This is compulsory, it include all necessary header file in your program, using **include** directive. | #include<stdio.h><br>#include<conio.h> |
| Definition Section: this section is used to define symbolic constant, using **define** directive. | #define PI 3.14 |
| Global Declaration Section<br>This section is used to declare a variable globally. Variable declared as a global can be accessed any where in the program. | int i;<br>float areaofcircle(float); |
| main() Section:<br>this is the heart of C Program from where actual execution of your program start.<br>This is compulsory , without main section C program cant run.<br>{<br>Declaration Part<br><br><br><br>Executable Part | void main()<br>{<br>//Declaration<br>Part float<br>r,area;<br><br>//Executable Part<br>scanf(“%f”,&r);<br>area=areaofcircle(r);<br>printf(“Area of Circle is :- %f“,area);<br>} |
| Subprogram Section (User Defined Section) this section is mandatory while we are dealing with user defined function.<br>Function1();Function2() | float areaofcircle(float r)<br>{<br>return PI * r * r;<br>} |

**2 Explain printf() and scanf() function with syntax.**

The stdio.h header file provides built in functions for reading (scanf) data from input devices (keyboard) and writing (printf) formatted data to output devices (monitor).

**scanf():**

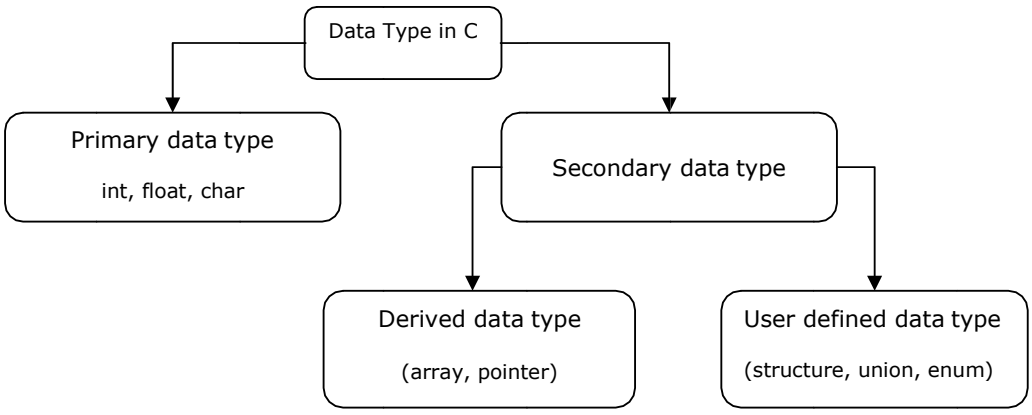
- scanf() is a library function that reads data with specified format from standard input (keyboard).
- Syntax: scanf(" Control string ", & variable name);
- Ex. For int a ; variable scanf () is as follow. scanf("%d",&a)
- First argument is the specification of format and other arguments are pointer variables to store data.
- scanf() stops when it exhausts its format string or when some input fails to match the control specification.

**printf():**

- Printf() is library function that prints or displayed data with format specified in printf statement.
- Syntax for printf is printf("Control String" , Variable name);
- For example int a=10; variable printf() is as follow. printf (" value of a is %d", a)

**3 Explain different data types available in C.**

A data type is a classification of various types of data, as floating-point, integer, or string. C is rich in its data types to allow programmer to select appropriate type of data type.




**1) Primary data types**

Primary data types are built in data types. They are directly supported by machine. They are also known as fundamental data types.

**a. int**

int is integer which is whole number without fraction part. Its range is machine dependent values.' C' has 3 classes of integer storage namely short int, int and long int. All of these data types have signed and unsigned forms.

| | signe | | unsigne | |
|---|---|---|---|---|
| | Size (bits | Range | Size (bits) | Range |
| short int | 8 | -128 to 127 | 8 | 0 to 255 |
| int | 16 | -32768 to 32767 | 16 | 0 to 65535 |
| long int | 32 | -2,14,74,83,648 to | 32 | 0 to |

**b. char**

char data type can store single character of alphabet or digit or special symbol. Each character is assigned some integer value which is known as ASCII values.

| | signe | | Unsign | |
|---|---|---|---|---|
| | Size (bits) | Range | Size (bits) | Range |
| char | 8 | -128 to 127 | 8 | 0 to 255 |

**c. float**

float data type can store floating point number which represents a real number with decimal point and fractional part. When the accuracy of the floating point number is insufficient, we can use the double to define the number. The double is same as float but with longer precision. To extend the precision further we can use long double which consumes 80 bits of memory space.

| | Size (bits) | Precision Digits | Range |
|---|---|---|---|
| float | 32 | 6 | 3.4E-38 to 3.4E+38 |
| Double | 64 | 14 | 1.7E-308 to 1.7E+308 |
| long float | 80 | | 3.4E-4932 to 1.1E+4932 |

**d. void**

The void type has no value therefore we cannot declare it as variable as we did in case of int or float or char. The void data type is used to indicate that function is not returning anything. It is default return type of 'C' language.

**2) Secondary data types**

Secondary data types are not directly supported by the machine. It is combination of primary data types to handle real life data in more convenient way. It can be further divided in two categories,

**a. Derived data type**

Derived data type is extension of primary data type. It is built-in system and its structure cannot be changed. Examples: Array, Pointer, etc…

i. Array: An array is a fixed-size sequenced collection of elements of the same data type.
ii. Pointer: Pointer is a special variable which contains memory address of another variable

**b. User defined data types**

User defined data type can be created by programmer using combination of primary data type and/or derived data type. Use can design it as per special requirements

i. structure: Structure is a collection of logically related data items of different data types grouped together and known by a single name.
ii. union: Union is like a structure, except that each element shares the common memory.
iii. enum: Enum is a user-defined type consisting of a set of named constants called enumerators. The enumerator names are usually identifiers that behave as constants in the language.

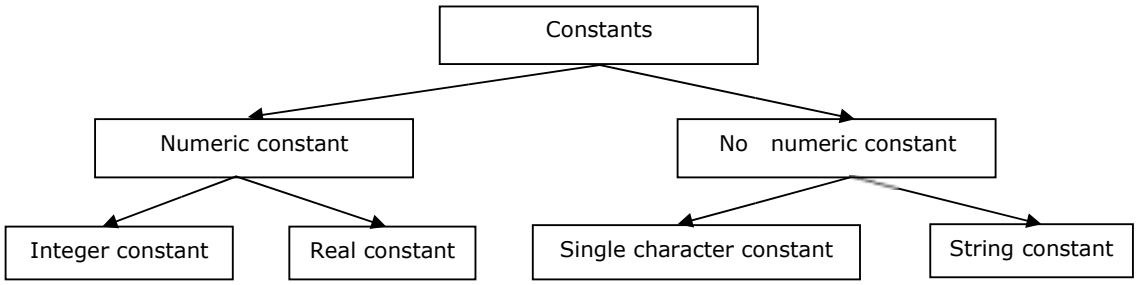
Syntax:
enum identifier {value1, value2,.... Value n};

```
#include<stdio.h>
#include<conio.h>
enum day {sunday = 1, tuesday, wednesday, thursday, friday, saturday};

void main()
{
   enum day d = thursday;
   printf("The day number stored in d is %d", d);
   getch();
}
```

**4 Explain types of constant in detail.**

Constant is something whose value does not change throughout the program.




**Integer constant:-**

- Integer constant is a number without decimal point and fractional part
- There are three types of integers constant.
  - Decimal integer
    Decimal integer consist of a set of digits, 0 to 9 having optional – or + sign. No other characters are allowed like space, commas, and non-digit charcters.
    Ex: 123, -321, 0, +78
  - Octal integer
    Octal integer consists of any combination of digits from the set 0 to 7.
    Octal numbers are always preceded by 0.
    Ex: 037, 0, 0551
  - Hexadecimal integer
    Hexadecimal integer consists of any combination of digits from the set 0 to 9 and A to F alphabets. It always starts with 0x or 0X. A represents 10, B represents 11… F represents 15. Ex: 0X2A, 0x95, 0xA47C.

**Real constant:-**

- The number containing the fractional part is called real number. Ex: 0.0083, -0.75, +247.0, -0.75.
- A real number may also be expressed in exponential notation.
- The general form is: mantissa e exponent, ex: 215.65 can be written as 2.1565e2.
- In exponential form, e2 means multiply by $10^2$.

**Single character constant:-**

- It contains single character enclosed within a pair of single quote mark.
- Ex: ‘5’, ‘A’, ‘;’, ‘ ‘

**String constant:-**

- A string constant is a sequence of characters enclosed within a double inverted comma.
- The characters may be le ter, number, special character, blank space, etc…
- Ex: “DIET”, “1988”, “?A.B,!”, “5+3”, etc…

- ‘A’ is charac er but “A” is string.

**5 What is variable? Give the rules to define variable name or identifier**

A variable is a data name that is used to store a data value. A variable may take different values at different times during execution of the program.

**Rules to define variable name:**

1. It must consists of only alphabets (a to z & A to Z), digits (0 to 9) & underscore (_).
2. First character must be an alphabet or an underscore.
3. Only first 31 characters are significant.
4. Cannot use C keyword.
5. Cannot contain space in naming for variable.

**Define the following terms:**

**Token**:

- Smallest unit of any language is known as token. The tokens are the basic building blocks which can be put together to construct programs.
- There are six classes of tokens in C: identifier, keywords, constants, string literals, operators and other separators

**Identifier**:

- An identifier is a sequence of letters and digits.
- The words which are defined by programmer in a program are known as identifier.
- Identifier refers to the name of variables, functions and arrays.

**Keyword**:

- Keywords are reserved words whose meaning is fixed and they are used for some special purpose by the compiler.
- They cannot be used by programmer for other purpose.

**void - data type**:

- void is a primary data type. void means nothing, no value.
- It is generally used to show that function is not returning anything.