UNIT-2

NOTES

1. What is Program?

A program is a set of instructions that a computer follows in order to perform a particular task.

2. What is Programming Language?

A programming language is a computer language that is used by programmers (developers) to communicate with computers. It is a set of instructions written in any specific language (C, C++, Java, Python) to perform a specific task.

3. Why do we use programming languages?

A programming language is mainly used to develop desktop applications, websites, and mobile applications.

4. Explain the types of Programming languages?

Types of programming language

1. Low-level programming language
2. High-level programming language
3. Middle-level programming language

1. Low-level programming language:

- low-level language, often known as a computer's native language, is a sort of programming language.
- It is very close to writing actual machine instructions, and it deals with a computer's hardware components and constraints.
- Assembly language and machine language are two examples of low-level programming languages.
- Machine language:-it is one of the most basic low level language. It is written in binary code or machine code, which means it basically comprises of only two digits – 1 and 0.

- Assembly language: This is the second generation programming language. It is a development on the machine language, where instead of using only numbers, we use English words, names and symbols.
  Example:- ADD R1,R2.

2. High-level programming language:

- A high-level language is a programming language that allows a programmer to create programs that are not dependent on the type of computer they are running on.
- Each instruction in high level programming is converted to machine language for the computer to comprehend.
- It helps faster program development, readable, well structured, and makes debugging easier.
- Execution speed of this language is much less compared to low level languages.

3. Middle-level programming language:

Programming languages with features of both Low Level and High-Level programming languages are referred to as “Middle Level” programming languages.
Example:- C, C++, and JAVA programming languages.

5. Give brief introduction about C language?

- The C Programming Language is a procedural, imperative, and general-purpose language utilized for coding applications that can run across multiple platforms.
- The C Language is developed by Dennis Ritchie for creating system applications that directly interact with the hardware devices such as drivers, kernels, etc.
- C programming is considered as the base for other programming languages, that is why it is known as mother language.
- The most of UNIX operating system are written in C language.

6. Explain the features or characteristics of C language?

1. Simple and Efficient:
The C Language is a simple language that is easy to learn even for a beginner and is super efficient to use both in terms of the time of development and time of execution.

2. Portability:
C Language program runs the same way everywhere. It means if you have written a simple C program like a program to find sum of N numbers in C, on your Windows operating system and then compiled it and run it, you can then take the compiled code and run it on any other operating system or machine, like, Linux or macOS, etc., your C program will always return the same result.

3. Structured Programming language:

- C language is a structured programming language because we can create functions in the C language.
- Using functions we can separate a particular operation from the main program and then use it again and again.
- A structured language is not just about having the ability to create functions, but it supports loops, conditional statements, etc.

4. Powerful:
It has a broad range of features like support for many data types, operators, keywords, etc., that allows the structuring of code using functions, loops, decision-making statements, etc.

5. Rich Standard Library:

- C supports various inbuilt functions and libraries that create development fast.
- The standard library support for the C language is superb and you will see that a lot of keywords or ready-made operations that you will use while writing code in C language are already pre-defined.
- These libraries are called Header files in C language.

6. Syntax-based Language:

There are proper rules for writing the code, and the C language strictly follows them.

7. Case-sensitive Language:

In C, the uppercase and lowercase characters are different. That means if is not the same as IF in C language.

8. Compiled Language:

The C language uses a Compiler to compile the code into **object code**, which is nothing but **machine code** that the computer understands. Hence to run a C language program we have to install a C language compiler first.

7. Write the applications of C language?

- Operating Systems
- Graphical User Interface
- New Programming Platforms
- Google
- Mozilla Firefox and Thunderbird
- MySQL
- Compiler Design
- Gaming and Animation etc.

8. Explain the basic structure of C program with an example?

Structure of C Program divides into Six Sections.

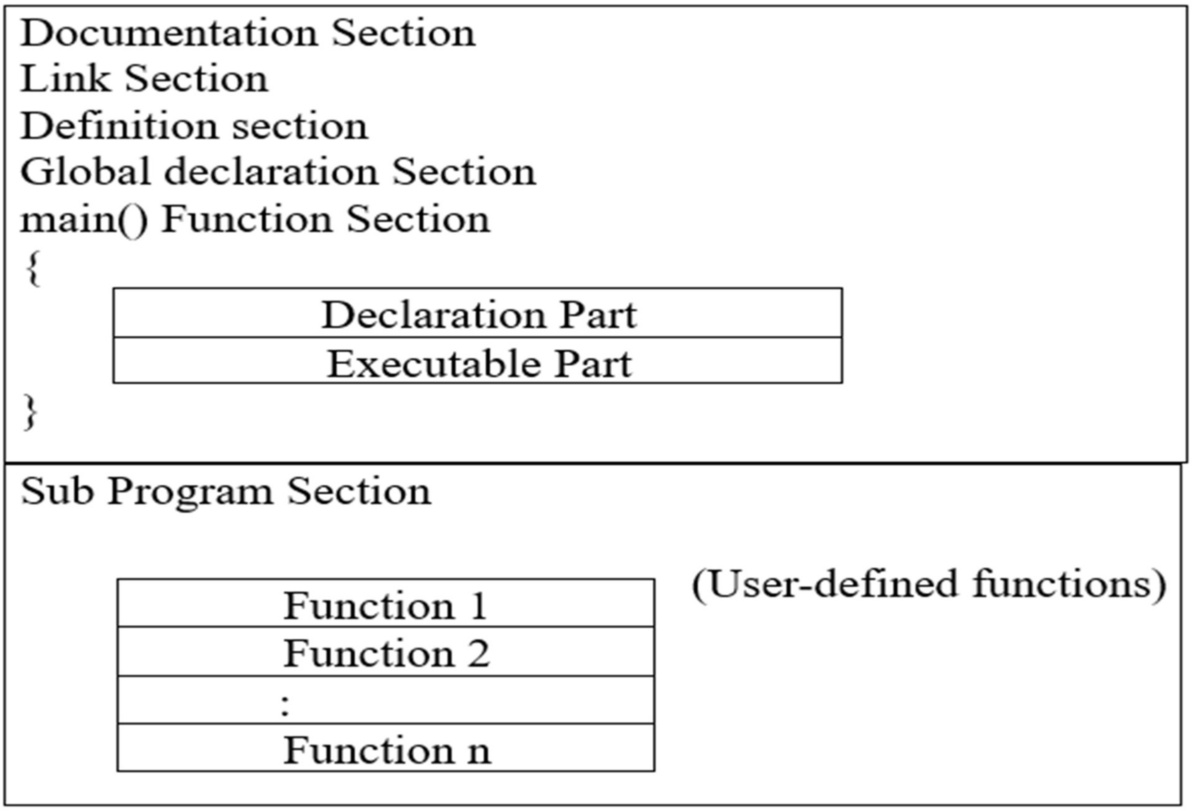
They are:-

- **Documentation Section:-** It is the section in which you can give comments to make the program more interactive. The compiler won't compile this and hence this portion would not be displayed on the output screen.

- **Preprocessor directives Section or Link Section:-** This section involves the use of header files that are to included necessarily program.

- **Definition section:-** This section involves the variable definition and declaration in C.

- **Global declaration Section:-** This section is used to define the global variables to be used in the programs, that means you can use these variables throughout the program.



Documentation Section
Link Section
Definition section
Global declaration Section
main() Function Section
{
Declaration Part
Executable Part
}
Sub Program Section
Function 1
Function 2
:
Function n
(User-defined functions)

- **Function prototype declaration section:-**
  This section gives the information about a function that includes, the data type or the return type, the parameters passed or the arguments.

- **Main function:-**
  It is the major section from where the execution of the program begins. The main section involves the declaration and executable section.

- **User-defined function section:-**

When you want to define your function that fulfills a particular requirement, you can define them in this section.

Example:- Program to demonstrate all the Sections of a C-Program

```
/* Program Name : Demo Program
    Programmer     : Shivaraj                 →    Documentation Section
    Date            : 04-09-20123
    Time            :  10 A.M.     */

#include<stdio.h>                             →         Link Section
#include<math.h>

#define MONTHS 12                             →      Definition Section
#define  AMOUNT 10000

Int a=10, b=20;                               →    Global declaration Section

main()                                        →    Main Function Section
{
   int x=30,y;                                →      Declaration part
   printf(“%d”, a);
   y= MONTHS – 2;
   printf(“%d”, y);                           →       Executable part
   MyFunction();
}

MyFunction()
{                                             →    Sub Program Section
  printf(“%d %d”,a,b);
}
```

9. Briefly explain about the Programming Style?

- All Statements should be in lowercase letters and upper case letters are only used for symbolic constants.
- Programmer can write the statements any where between the two braces following declaration part.
- The user can also write more than one statement in one line separating them with a semicolon.

```
Example:-  a= b + c;
           d= e * f;
              or
           a=b + c; d= e * f;
```

- The opening and closing braces should be balanced.
- Use comments whenever necessary. Comments increases the program readability and also help to understand the logic.

10. What is Modular Programming? Mention any two advantages of it?

Modular Programming is a program design technique in which a large program is divided into sub programs/functions that are called modules.

Advantages:-

- It improves maintainability of a program.
- It makes software development , debug, modify, update faster and easy.

11. Explain the process of Creating and Executing a 'C'- Program?

There are totally six stages in this process:-

`1. Creating a C Program:-

- A program can be written in any text editor like notepad.(But we are using turbo C++ software).
- After writing a program, it must saved in disk with the extension ".c".
- C program file has extension .c(Example:- hello.c , program.c).
- Source file will be generated in this stage.

2. Preprocessing:-

- The C preprocessor obeys special commands called preprocessor directives, that begin with #(known as directives).
- Preprocessor will look for the header files used in program such as
- #include<stdio.h> and copy the header files into the source code.

3. Compiling:-

- The Compiler converts the source code(C program) into machine language code(also called as Object code or object file).
- If any syntax errors in the source code, then compiler checks for it.
- Once all the errors are corrected then compiler converts source code to machine code.(example:- hello.obj).

4. Linking:-

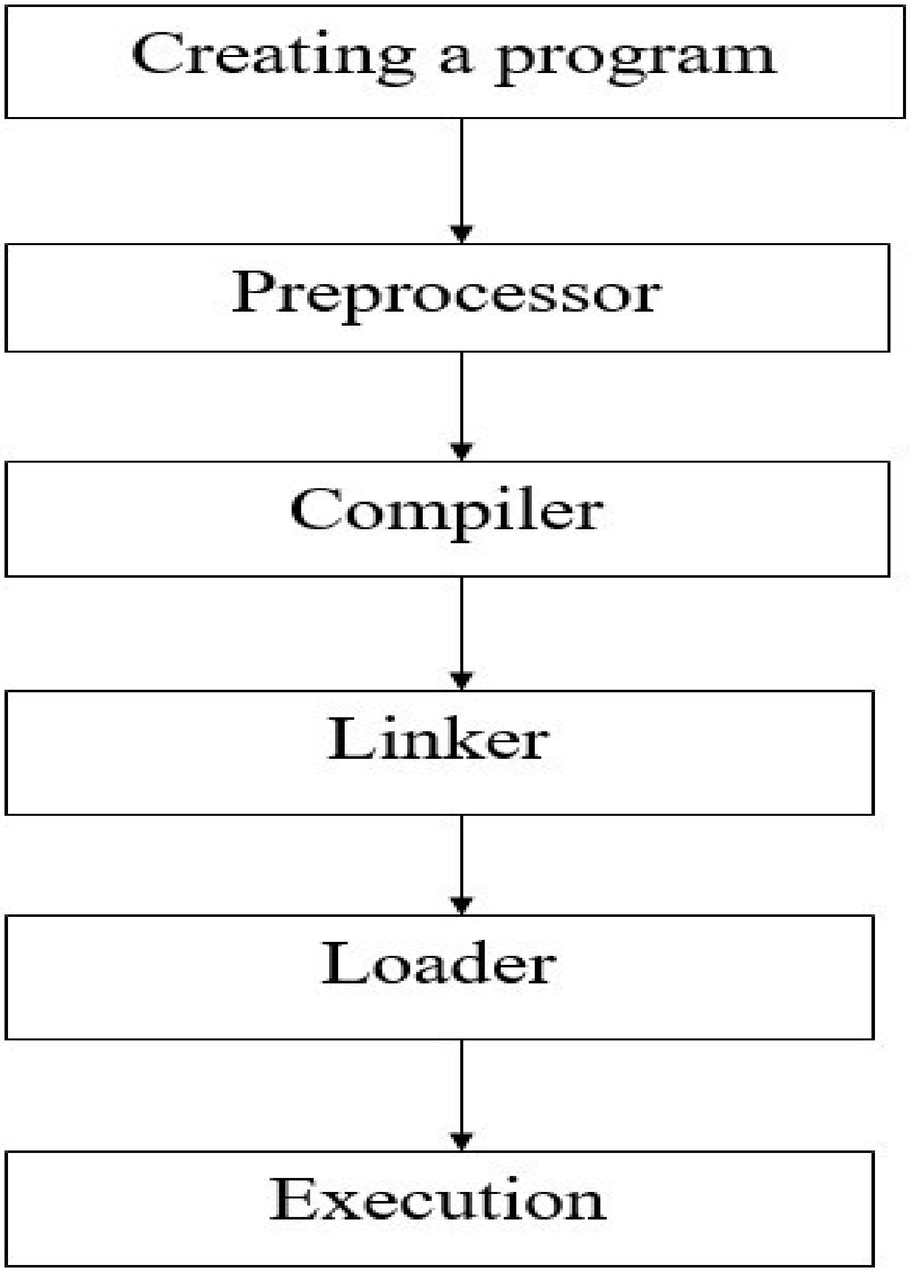
- Object file are not executable file so in order to make executable file we use linker.
- The linker links the program with external library files which contains the code for predefined functions and creates an executable file.
- The extension of the executable file is ".exe"(for example: hello.exe).

5. Loading:-

- Before a program can be executed, the program must be placed in memory. This
- is done by the loader, which takes the executable file from disk and transfers it to memory.
- The loader will load the .exe file in RAM and inform the CPU with the starting point of the address where this program is loaded.

6. Execution:-

- Once the executable object code is loaded into memory, CPU reads each instruction and executes it.
- We need to test to determine whether programs work properly or not.
- If program does not work properly we need to return to stage 1 for modification.



The process of Creating and Executing a ‘C’- Program

12. Explain the Character Set in C?

The Character Set refers to a set of all the valid characters that we can use in the source program for forming words, expressions, and numbers.

- C language supports a total of 256 characters.
- Every C program contains statements. These statements are constructed using words and these words are constructed using characters from C character set.
- C language character set contains the following set of characters...
- ➢ Alphabets
- ➢ Digits
- ➢ Special Symbols

Alphabets

- C language supports all the alphabets from the English language. Lower and upper case letters together support 52 alphabets.
- lower case letters - **a to z**
- UPPER CASE LETTERS - **A to Z**

Digits

- C language supports 10 digits which are used to construct numerical values in C language.
- Digits - **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

Special Symbols

- C language supports a rich set of special symbols that include symbols to perform mathematical operations, to check conditions, white spaces, backspaces, and other special symbols.
- Special Symbols - **~ @ # $ % ^ & * ( ) _ - + = { } [ ] ; : ' " / ? . > , < \ | tab newline space NULL bell backspace vertical tab etc.,**
- Every character in C language has its equivalent ASCII (American Standard Code for Information Interchange) value.

13. What are C Tokens? Mention its types?

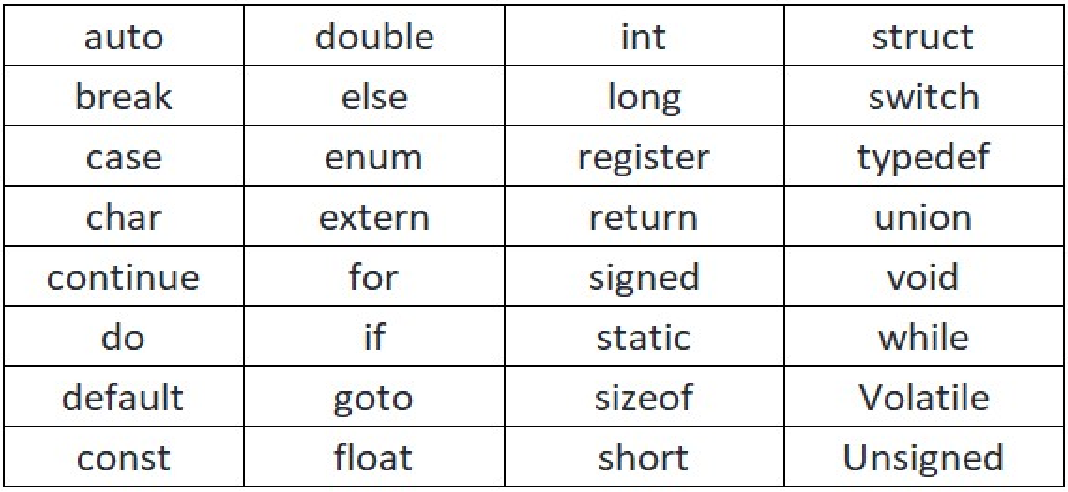
One or more characters grouped together to form basic elements of C are known as C Tokens.

Types of Tokens in C:-

- Keywords
- Identifiers
- Constants
- Strings
- Special Symbols
- Operators

14. What are keywords? Explain.

- The keywords are reserved for specific meanings in C language.
- When keywords are combined with the formal syntax, it forms the C programming language.
- **C** language supports **32** keywords.



auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
continue	for	signed	void
do	if	static	while
default	goto	sizeof	Volatile
const	float	short	Unsigned

15. What are Identifiers? Explain.

- Identifier is a name given to variable , function or any other user defined item.
- Identifiers must be unique. They are created to give a unique name to an entity to identify it during the execution of the program.

- For example:

int money;
double accountBalance;
Here, money and accountBalance are identifiers.

Rules for Identifiers:-

- They must begin with a letter or underscore(_).
- They must consist of only letters, digits, or underscore. No other special character is allowed.
- It should not be a keyword.
- It should be a single word. i.e., No blank space is allowed.
- Identifiers are case-sensitive.
- The maximum length of an identifier is 31 characters.

Valid Identifiers	**Invalid Identifiers**
ADD	3add
Add	Add+123
_Add	for
Student_Name	ABC XYZ
Name	Student%123
A312B	_Add*5Add

16. Explain the Special Symbols used in C?

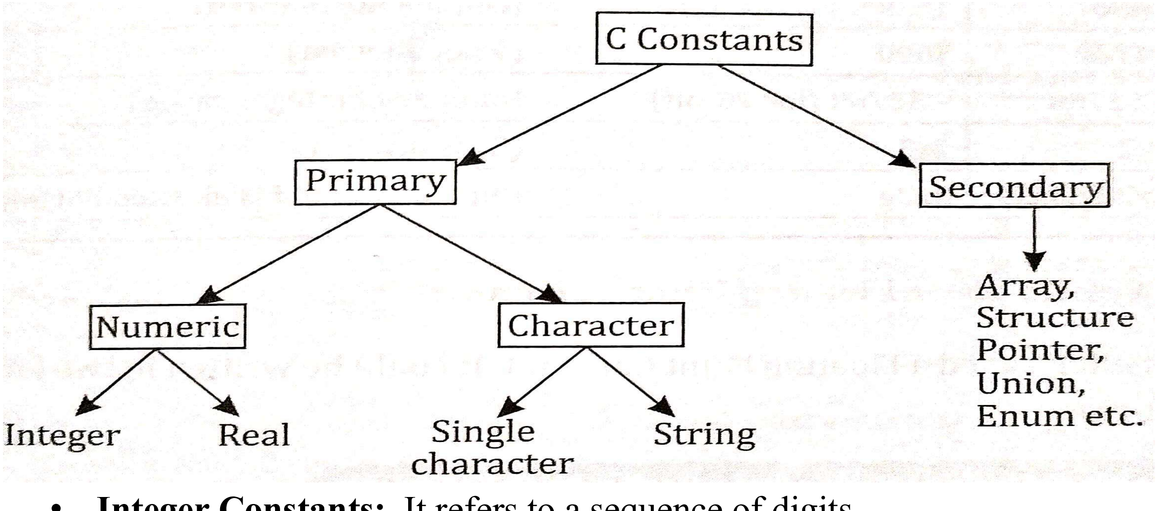
The following special symbols are used in C having some special meaning and thus, cannot be used for some other purpose. Some of these are listed below:

- Brackets[]: Opening and closing brackets are used as array element references. These indicate single and multidimensional subscripts.
- Parentheses(): These special symbols are used to indicate function calls and function parameters.
- Braces{}: These opening and ending curly braces mark the start and end of a block of code containing more than one executable statement.
- Comma (,): It is used to separate more than one statement like for separating parameters in function calls.
- Colon(:): It is an operator that essentially invokes something called an initialization list.

- Semicolon(;): It is known as a statement terminator. It indicates the end of one logical entity. That's why each individual statement must be ended with a semicolon.
- Asterisk (*): It is used to create a pointer variable and for the multiplication of variables.
- Assignment operator(=): It is used to assign values and for logical operation validation.
- Pre-processor (#): The preprocessor is a macro processor that is used automatically by the compiler to transform your program before actual compilation.
- Period (.): Used to access members of a structure or union.
- Tilde(~): Used as a destructor to free some space from memory.

17. What are Constants? Explain.

- Constants in C refer to the fixed values that do not change during the execution of a program.   OR
- A Constant is a token with fixed value that does not change.
- There are two types of constants in C: Primary and Secondary Constants.



- **Integer Constants:** It refers to a sequence of digits.

Example:- 650 , +666, -100, -32000

- **Real Constants :** it is often called a Floating point constant.

Example:- 426.00, -66.6666, +.5

- **Character Constants:** It consists of a single character, single digit, or a single special enclosed within a pair of single inverted commas. The maximum length of a character is one character.

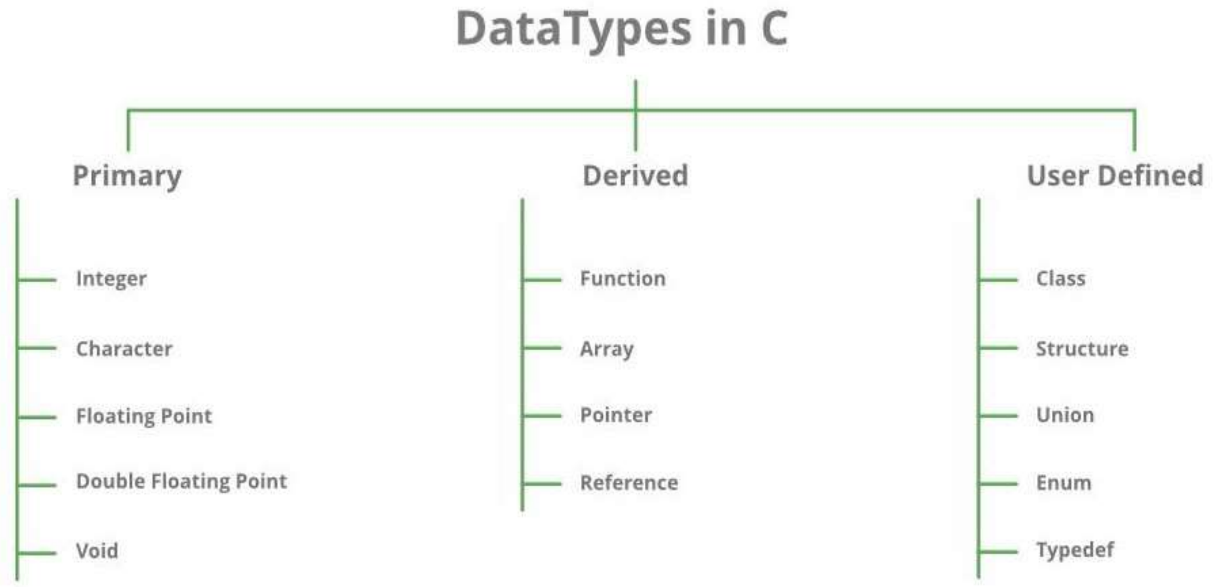
Example:- ‘a’ is a character constant.

- **String Constants:** It is a sequence of one or more characters enclosed within a pair of double quotes. The characters may be letters, numbers, special characters and blank spaces.

Example:- “Hello”

18. What are data types? Explain.

It specifies the type of data that the variable can store like integer, character, floating, double, etc.



Primary Data Types:-

1)Integer Data Type

- The integer datatype in C is used to store the whole numbers without decimal values. Octal values, hexadecimal values, and decimal values can be stored in int data type in C.
- **Range:** -2,147,483,648 to 2,147,483,647
- **Size:** 4 bytes
- **Format Specifier:** %d
- **Syntax of Integer**

We use int keyword to declare the integer variable:

int var_name;

Example:- int sum, j;
int a=10, b=20, result;

The integer data type can also be used as

- **unsigned int:** Unsigned int data type in C is used to store the data values from zero to positive numbers but it can't store negative values like signed int.
- **short int:** It is lesser in size than the int by 2 bytes so can only store values from –32,768 to 32,767.
- **long int:** Larger version of the int data type so can store values greater than int.
- **unsigned short int:** Similar in relationship with short int as unsigned int with int.

2)Character Data Type

- Character data type allows its variable to store only a single character. The size of the character is 1 byte. It is the most basic data type in C.
- It stores a single character and requires a single byte of memory in almost all compilers.
- **Range:** (-128 to 127) or (0 to 255)
- **Size:** 1 byte
- **Format Specifier:** %c
- **Syntax of char**

The **char keyword** is used to declare the variable of character type:

char var_name;

Example:- char ch , j, str;
Char ch= 'A', i= 'B', str='C';

3)Float Data Type

- In C programming float data type is used to store floating-point values.
- Float in C is used to store decimal and exponential values.
- It is used to store decimal numbers (numbers with floating point values) with single precision.

- **Range:** 1.2E-38 to 3.4E+38
- **Size:** 4 bytes
- **Format Specifier:** %f
- **Syntax of float**

The **float keyword** is used to declare the variable as a floating point:
float var_name*;*
Example:- float a,b,c=3.142;

4)Double Data Type

- A Double data type in C is used to store decimal numbers (numbers with floating point values) with double precision.
- It is used to define numeric values which hold numbers with decimal values in C.
- It can easily accommodate about 16 to 17 digits after or before a decimal point.
- **Range:** 1.7E-308 to 1.7E+308
- **Size:** 8 bytes
- **Format Specifier:** %lf
- **Syntax of Double**

The variable can be declared as double precision floating point using the **double keyword.**
double var_name*;*
Example:- double d1,d2,d3 = 1.5e55;

5)Void Data Type

- The void data type in C is used to specify that no value is present.
- It does not provide a result value to its caller.
- It has no values and no operations.
- It is used to represent nothing.
- Void is used in multiple ways as function return type, function arguments as void, and pointers to void.

19. What are Variables? Explain.

A Variable is a named location in memory that is used to hold a value that can be modified by the program.

Syntax for using Variable:-

data_type var_name;

where, data_type is a valid data type (along with datatype modifiers, if required) and var_name is the name of the variable.

Example:- int marks;

marks is the name of the variable, and it can store values of int type.

OR

datatype variable_list;

where, data_type is a valid data type (along with datatype modifiers, if required) variable_list refers to one or more identifiers names separated by commas.

Example:- int i,j,sum;

Where i,j,sum are called variables of type int.

Assigning value to Variable:-

```
// variable declaration
    int marks;
 // variable definition
     marks = 10;
int marks = 10;    // variable declaration and variable definition in one
step
```

Rules to Name a Variable:-

- When you create a variable, you should always give a meaningful name to the variable. And follow the below rules for naming the variable:
- Variable names must not start with a digit.
- The variable name can consist of alphabets, digits, and special symbols like underscore _.
- Blank or spaces are not allowed in the variable name.
- Keywords are not allowed as a variable name.

- Upper and lowercase names are treated as different, as C is case-sensitive, so it is suggested to keep the variable names in lowercase.

Valid Variable Names	Invalid Variable Names
Sum	for
Name	123Add
_Add	Accno%123
average_score _	Average score

20. What are Local variables and Global variables?

- The variables declared inside the functions are called Local variables.
- The variables declared outside of all the functions are called Global variables.

Example:-

```
int a , b, c;          →          global variables
void main()
{
   int i, j , sum;       →          local variables
-----------------
------------
}
```

21. What are the difference between Identifiers and Variables?

Identifiers	Variables
It is used to name a variable, a function, a class, a structure, a union.	It is used to give a name to a memory location that holds a value.
Identifiers are created assign a name to an entity.	Variable is created to assign a unique name to a specific memory location.
All identifiers are not variables.	All the variables names are identifiers.
Identifier can take more number of characters.	Variable takes less number of characters.