1. Why C is called middle level language?

- C combines features of higher level language and lower level language that's why it is known as middle level language.

Features of higher level language

- C tokens and keywords are like natural English language.
- C supports functions for structured programming.
- C supports variety of data types which makes programming efficient and easy.
- C is easy to learn, write applications, debug and modify program.

Features of lower level language

- We can write assembly language statements in C program.
- Using pointer, we can directly access any memory location.
- Bit wise operations are very much useful in system programming.

2. Give the advantages & disadvantages of C language or Features of C language.

- ***ADVANTAGES:***
- C is stable language: C is 35 years old language and it is not revised since last 20 years.
- C is modular Language: Large program can be divided into smaller functions to reduce complexity.
- C is portable language: Program written on one computer can be executed on other computer without any modification.
- C is concise language: There are only 32 keywords in C.
- C data types: C supports variety of data types for efficient and easy programming.
- C operators: C supports most of the possible operators for variety of operations.
- Speedy Execution: Execution of C program is much faster than languages program.
- Core language: We can write application software as well as system software in C. Also it is base of many other programming languages.
- ***DISADVANTAGES:***
- C doesn't concept of Namespace.
- C doesn't have the concept of Constructors and Destructors.
- It does not support the object oriented concept such as polymorphism, data hiding and the others.
- It is case sensitive so mixing case makes difficult while programming
- There is not enough library functions for today's programming strategy.

3. Draw and explain Structure of C Program

Structure of a C program	Example of a C program.

Documentation Area
Link area
Definition Area
Global Deceleration Area
 	
main()
{
Deceleration Area
Executable Area
 	
}
 	
Sub Program Area
)User Defined Area(
Function1()
Function2()
	// This program is to find area of circle
	#include<stdio.h>
	#define PI 3.14
	int i;
	float areaofcircle(float);
	void main()
	{ //Declaration Part
	float r,area;
	//Executable Part
	scanf ("%f",&r);
	area=areaofcircle(r);
	printf("Area of Circle is :- %f",area);
	}
	float areaofcircle(float r)
	{
	return PI *r*r;
	}

Documentation Section The documentation section consists of a set of comment lines giving the name of the program, the author and other details, which the programmer would like to use later.

Link Section The link section provides instructions to the compiler to link functions from the system library.

Definition Section This section defines all the symbolic constants.

Global Declaration Section This section is used to declare the global variables used in the program. Global variables are the variables used in more than one function. This section also declares all the user defined functions.

Main() Function Section

Every C program must have one main function section. This section contains two parts:

1) **Declaration Part** declares all the variables used in the executable part.

2) **Execution Part** must contain at least one statement which contains instructions to perform certain tasks.

The declaration and executable part must appear between the opening and closing braces. All statements in the declaration part must end with the semicolon.

Subprogram Section

This section contains all the user defined functions that are called in the main function. These functions are performed by user specific tasks and this also contains set of program statements. They may be written before or after a main () function and called within main () function.

C Comments

- A comment is a line or paragraph of text in a program which is used for documentation or message purpose.
- Comment is ignored by the compiler when it compiles programs.
- The C/C++ language accepts two types of comments:
    1. To write a comment on one line, type two forward slashes // and type the comment. Anything on the right side of both forward slashes would not be read by the compiler
       e.g.: // This program calculates area of circle. This is written by XYZ.
    2. To write a multi line comment, you can start it with a forward slash / followed by an asterisk *, write the comment. To end the comment, type an asterisk * followed by a forward slash /. This type of text can be spread on various lines

  e.g.: /* This program calculates area of circle. This is written by XYZ.
  This is written on 1-Jan-2011.*/

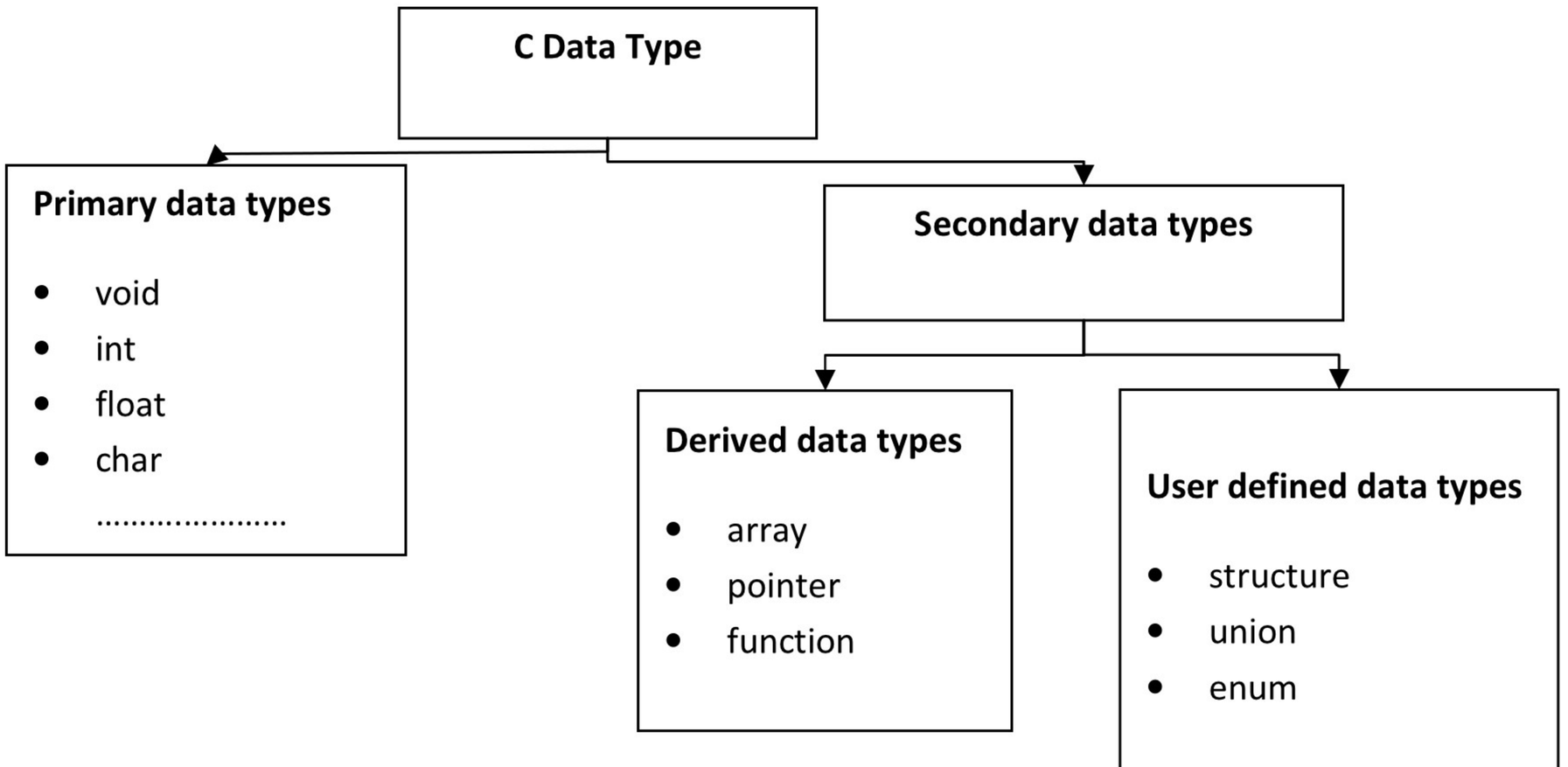
C Header files

- In C programming, a source code file that contains common functions and types that all programmers may use as required is called header file.
- We can include them in any program using #include statement.
- It is also known as include files or library files.
- Header files use .h extension.
- All functions are grouped category-wise and stored in different files.
  stdio.h : Standard Input/ Output
  math.h : Mathematical functions
  E.g-.#include<stdio.h>
  -#include<conio.h>

4. Explain various C data types

- Data types are used to store various types of data that is processed by program.
- Based on data type, memory is allocated to the variable to store data.

- We should select appropriate data type based on our data and its operations.
- C language is richset of data types, it can be mainly classified in three categories,



Primary data types

- Primary data types are built in data types.
- Primary data types are machine specific data types.
- Programmers mostly use these data types.

1. Integer numbers

- Integer numbers are whole numbers without decimal point and fractional part, e.g. 20, -154, 0, 77, etc…
- Signed number can store positive or negative values, e.g. 20, -154. By default variable is signed.
- Unsigned number means value can be only zero or positive, e.g. 20, 100, 0.

Size (In Byte)		
Data Type	Signed	Unsigned
short int	1 Byte (Range -128 to127)	1 Byte(Range -0 to255)
Int	2 Byte (Range -32768 to32767)	2 Byte (Range -0 to65535)
long int	4 Byte	4 Byte

2. Float or real numbers

- Float numbers contain decimal point and fractional part, e.g. 3.25, 7.89, -100.25, 10.00
- Precision means with how much accuracy we can store digits after decimal points. e.g. 3 then we can operate on 3 digits after decimal point.

Data Type	Size(In Byte)	Precision
Float	4	6
Double	8	14
long double	10	

3. Character data type

- Character type variable can hold a single character, e.g. ‘A’, ‘z’, ‘9’, ‘ ‘
- Internally, each character is assigned some integer value known as ASCII values.

Size (in Byte)		
Data Type	Signed	Unsigned
Char	1 Byte (Range-128 to127)	1Byte(Range-0 to255)

4. void data type

- void means nothing, no value. So we can not declare variable of type void.
- void is used to indicate that function is not returning any value.

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 by compiler.
- It is built-in in the C language and its structure cannot be changed.

i. Array: An array is a fixed-size sequenced collection of elements of the same data type grouped under a single variable name.
ii. Pointer: Pointer is a special variable which stores address of another variable.
iii. Function: Function is a block of code for some particular task.

b. User defined data types

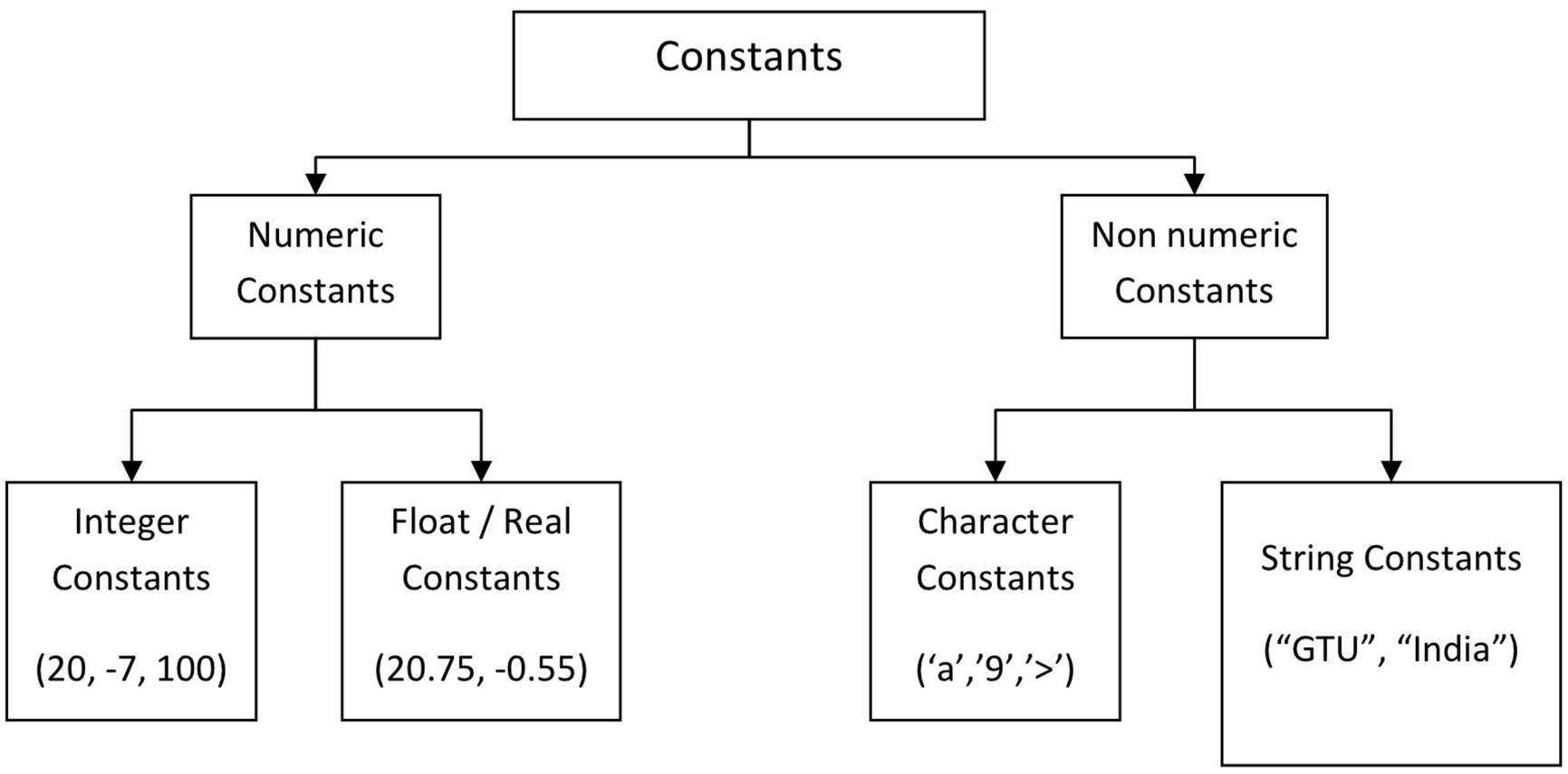
- User defined data type is created by programmer as per requirement.
- It is combination of primary data type and/or derived data type.
- It helps to organize and handle real world data in more convenient way.

i. Structure: structure is a collection of logically related data items of different data types grouped under 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 identifiers that behave as.
Example:

```
enum day {Mon, Tue, Wed, Thu, Fri, Sat, Sun};
enum day week1stday ;
week1stday = Mon;
```

5. Explain C constants in detail.

- Constants are values that never change.
- A "constant" is a number, character, or character string that can be used as a value in a program.



Integer constant:

- Integer constant is a constant number without fractional parts.
- There are 3 types of integer constant
- Decimal Integer: It consists of a set of digits (0 to 9), having optional + or – sign. No other character is allowed. E.g. 123, +78, -120

- Octal integer: It consists of any combination of digits from the set 0 to 7 and + or - sign. Octal numbers are always preceded by 0 (Zero). E.g. 037, -0551
- Hexadecimal integer: It 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. E.g. 0X2A, 0x95, 0xA47C

Float or 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

Character Constant

- It contains single character enclosed within a pair of single quote mark.
- E.g. 'a', 'Z', '+', ' '

String Constant

- A string constant is a sequence of characters enclosed within a double inverted comma.
- The characters may be letter, number, special character, blank space, etc…
- E.g. "GTU", "1988", "?A.B,!", "5+3", etc…

'A' is character but "A" is string.

6. What is variable? What are the rules to declare variable names?

- A variable is a name used to refer to some location in memory – a location that holds a value.
- A variable may take different values at different times during execution of the program.
- At the time of variable declaration, we have to tell which type of data it can hold.

Rules to define variable name:

- It must consists of only alphabets (a to z & A to Z), digits (0 to 9) & underscore (_).
- First character must be an alphabet or an underscore.
- Only first 31 characters are significant.
- Cannot use C keyword.

7. Define keyword and identifier. Gives examples and write the rules for declaring identifiers.

Keywords:

- Keywords are the words whose meaning has already been fixed to the C compiler (or in a broad sense to the computer). The keywords cannot be used as variable names because if we do so we are trying to assign a new meaning to the keyword, which is not allowed by the computer.
- All keywords have fixed meanings and these meanings cannot be changed. Keywords serve as basic building blocks for computer statements.
- There are only 32 keywords available in ANSI C.

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

Identifiers:

- Identifiers are used as the general terminology for the names of variables, functions and arrays. These are user defined names consisting of sequence of letters and digits with either a letter or the underscore (_) as a first character.
- Both uppercase and lowercase letters are permitted, although lowercase letters are commonly used.

Write the rules for Identifiers:

1. The first character in an identifier must be an alphabet or an underscore.
2. Must contain only letters, digits or underscores.
3. Uppercase and lowercase letters are distinct. That is, identifiers are case sensitive.
4. Special characters, blank spaces or keywords are not allowed within an identifier.
5. Only first 31 characters are significant.

8. Explain backslash and trigraph characters.

Backslash Characters

- Back slash constants are a special type of character constant which are consists of two characters.
- This is known as escape sequence. Escape sequence starts with backslash '\' character.

Escape sequence	Meaning
'\0'	End of string – NULL.
'\n'	End of line – takes the control to next line.
'\r'	Carriage return – takes the control to the next paragraph.
'\f'	Form feed- takes control to the next page.
'\t'	Horizontal tab
'\b'	Back space
'\\'	Prints backslash character \
'\a'	Alert – provide edible alert.

Trigraph Characters

- "Trigraph" sequence provides a way to enter certain characters which are not available in some keyboard of non English language.
- Each trigraph character consists of a three characters. First two characters are ??and third character determines translation character.

Trigraph sequence	Translation
??=	# number sign
??(	[left bracket
??)	] right bracket
??<	{ left bracket
??<	} right bracket
??!	| vertical bar
??/	\ backslash

9. Explain Constant and Volatile variable.

Constant

- When we want to use certain variables to remain constant during the execution of a program, we may declare the variable with the qualifier *const* at the time of initialization. *const* means that something is not modifiable, so a data object that is declared with const as a part of its type specification must not be assigned another value in any way during the execution of the program.
- For Examples:

```
const int five = 5;
const double pi = 3.141593;
```

const objects may not be changed Thus the following are illegal:

```
const int five = 5;
const double pi = 3.141593;
pi = 3.2;
five = 6;
```

Volatile

- In "C", if you are declaring any variable then by default that variable can be known as volatile variable. Volatile means those variables whose value can be modify after declaration.

For Example:

```
volatile int date;
```

10. What is a control string? Write the symbols and meaning of it.

- As we know in 'C' language to access different kind of datatypes, we have to use different-different keywords as well as different memory allocation is there.
- So like that to access different kind of datatypes, we have to use different kind of control strings, which are as below;

Data types	Control String
signed char / unsigned char	%c
short signed int / signed int	%d
short unsigned int / unsigned int	%u
long signed int	%ld
long unsigned int	%lu
float	%f
double	%lf
long double	%Lf