*[Short Questions]*

1. **Define Program**
   - A well-defined set of instructions given to the computer to solve a problem is called **computer program.** Program is also called **software.**
   - **Program or software** is written in any programming language.
2. **What is Programming Language**
   - Set of words and symbols used to write a computer programs is called programming language.
   - It is a mean of communication between user and computer.
   - Programming languages are used to write computer programs or software's
3. **What is Low level language**
   - A type of language that  is near to computer hardware and far from human language
   - Two type of low-level languages are **i)** Machine language **ii)** Assembly language
4. **What is Machine Language**
   - A type of language in which instructions are written in binary form i.e (0,1)
   - It is the only language that is directly understood by the computer
   - It is the native language of computer
   - Programs written in machine language are **Machine-dependent.**
5. **What is Assembly Language**
   - It is a Low level language ,it is one step higher than machine language
   - Machine instructions are replaced with English –like words known as **Mnemonics** (Ne-monics)
6. **What is Assembler**
   - Assembler is a translator that is used to translate the instructions of Assembly language into machine language
7. **What is high level language**
   - A type of language that is close to human language is called high-level language.
   - Instructions in these languages are similar to English language
   - These languages are easy to understand
8. **What is Syntax**
   - Collection of Rules for writing programs in a programming Language is known as Syntax.
   - Each language has its own syntax.
   - All program statements are written according to the syntax of a language.
9. **What is Machine dependent**
   - A program written in machine language is Machine-dependent.
   - Machine dependent programs are executed on the computers for which these programs are written.
10. **What is Machine Independence**
    - High-level languages provide machine independence. It means that programs written in high-level language can be executed on different types of computers.
    - **Example**: Program written in C can be executed on Intel Processors and Motorola Processors
11. **What is Source Code**
    - A program written in a high-level language is called source code. Source code is also called source program.
    - Source code cannot be executed by computer directly
    - Source code in converted in object code and then executed.
12. **What is Object Code**
    - A program written in a Machine language is called Object code. Object code is also called Object program.
    - Computer understand object code directly
13. **What is Compiler**
    - A Compiler is a program that converts the instruction of a high level language into machine language **as a whole**.
    - Compiler also checks each statement in the source program and generates machine instruction.
    - Program containing an error cannot be compiled.
    - A new file with **.obj extension** is created if the process of compilation is successful
14. **What is Interpreter**
    - An interpreter is a program that converts one statement of a program into machine at one time
    - Error is Displayed during program development is an **advantage of Interpreter**.
    - **Disadvantage** of interpreter is that it is not very efficient.

15. **History of C Language**
    - C Language was developed by **Dennis Ritchie** in 1972 at Bell Labs.
    - C language was derived from an earlier programming language called B.
    - B Language was developed by **Ken Thompson** in 1969-70.
16. **What is ANSI C**
    - The earlier version of C was Known as **K& R** (Kernighan and Ritchie) C . The American National Standard Institute (ANSI ) developed a standard version of the language .
    - The Standard version is known a **ANSI C**
17. **What is Turbo C++ or What is an IDE**
    - **Compiler** used for C language is called Turbo C++
    - It is implementation of Borland International for C Language.
    - It is an Integrated Development Environment **(IDE)** known as **TC Editor.**
    - **IDE** is Used to Create, save, compile and execute C Programs
18. **What is Linker**
    - A Program that combines the **object Program** with **additional library files** is known as Linker.
    - It is part if C Compiler. It is used to perform the process of **Linking.**
    - A new file with **.Exe extension** is created if the process of linking is successful
19. **What is Loader**
    - A program that places the executable file in the memory is known as loader.
    - Program must be loaded in the memory in order to execute it.
    - A program can be loaded into memory when we execute our program
20. **What is preprocessor Directive**
    - Preprocessor Directive is an **instruction given to the compiler** before the execution of program
    - Preprocessor directive is also known as **compiler directive**
    - Preprocessor directives are processed by the program known as **preprocessor**
    - It modifies the C Source Program before compilation
21. **What are Types of Preprocessor Directives**
    - Two Types of Preprocessor directives are
        - include Preprocessor
        - define Preprocessor
22. **What is include Preprocessor Directive**
    - The include preprocessor directive is used to include Header files in the program
    - It enables a program to access a library files for different tasks such as input/output

| | |
|---|---|
| **Syntax**<br>#include < Headerfile > | **Example**<br>#include <stdio.h><br>#include <math.h> |

23. **What is define Preprocessor Directive**
    - The define directive is used to define a constant in the program
    - It starts with the symbol # and it is not terminated with semicolon

| | |
|---|---|
| **Syntax**<br>#define identifier value | **Example**<br>#define PI 3.141593 |

24. **What is header file**
    - Header files are collection of standard library functions to perform different tasks.
    - There are many header files for different purposes.
    - Each header file contains different types of predefined functions.

| | |
|---|---|
| **Syntax**<br>#include < Headerfile > | **Example**<br>**#include <stdio.h>**<br>**#include <math.h>** |

25. **What is main() function**
    - main() function is the place where the execution of C program starts
    - when the program is executed, control enters main() function and starts executing its statement
    - program must contain main() function
    - program is compiled but cannot be executed without main function
26. **What is bug and debugging**
    - Programmer can make different **errors** while writing programs
    - Error in the program is called **bug**
    - Process of finding and removing bugs in known as **debugging**

### 27. What are delimiters?

- Statements of C program are written in curly braces. The curly brace { is called opening and } is called closing brace.
- The braces { } are called delimiters
- These statements are known as body of program

### 28. What is Syntax error

- Syntax error is a type of error that occurs when an **invalid statement** is written in program
- Compiler detects syntax errors and display message that describe the cause of error
  - **Causes of Syntax error**
    - Statement terminator is missing
    - Misspelled keyword
    - Any of the delimiters is missing
  - **Example of Syntax error**
    - Typing **viod** instead of **void**

### 29. What is logical error

- A type of error that occurs due to poor logic of the programmer
- A statement with logical error may produce unexpected and wrong results
- Logical errors are difficult to find ,programmer needs to review the whole program

### 30. What is run time error

- A type of error that occurs during the execution of program is known as run-time error
- Run-time errors normally occur due to wrong input from the user
- Run-time error is detected and displayed during execution
  - **Example**
    - Dividing a number by zero produce run-time error

### 31. What is structured programming languages

- Entire logic of the program is divided into a number of smaller modules or functions
- Each module is a piece of code that implements a different functionality
- Programs written in these languages are easy to understand, modify and debug

### 32. What is unstructured programming

- Entire logic of the program is implemented in a single module or function
- Programs written in these languages are error prone ,difficult to understand, modify and debug

### 33. Types of files in c programming

| **File Extension** | **File Types** | **Created after/by** |
|---|---|---|
| .h | header file | available in INCLUDE Directory (created during installation of Turbo C++) |
| .c | Source Code | Created by program |
| .obj | object code | created after compilation |
| .exe | executable file | created after linking |

### 34. First C Program

```
#include<stdio.h>
void main(void)
{
printf(“Hello Word”);
}
```

### 35. Characteristics of high level languages

| | |
|---|---|
| 1. Easy to learn | **2.** Easy error detection |
| 3. Standardized Syntax | 4. Deep hardware knowledge not required |
| 5. Machine independence | 6. More programmer |
| 7. Shorter program | |

### 36. Advantages of C language

| | |
|---|---|
| 1. Convenient language | 2. Well-structured language |
| 3. Machine independence | 4. Modularity |
| 5. Case sensitivity | 6. Hardware control |
| 7. Small language | 8. Fast code generation |

### 37. What necessary steps are taken to prepare C Program for execution?

| | | |
|---|---|---|
| 1. Creating and editing | 2. Saving | 3. Compiling |
| 4. Linking | 5. Loading | 6. Executing |