

SHORT QUESTIONS

Q.1 Define Computer.

Ans. Computer is an electronic device that accepts data process it and produce information according to the instructions given to it.

Q.2 Define Data.

Ans. The collection of raw facts and figures is called data.

Q.3 Define Information.

Ans. The organized and meaningful form of data after processing is called information.

Q.4 Define Program / Software.

Ans. A set of instructions given to the computer to solve any problem is called a program. A program is written in a computer language.

Q.5 Define programming Language

Ans. It is the way of communication between user and computer. All the programs are written in computer language.

Q.6 Define High level language

Ans. It is closed to human language. User can learn and understand high-level language easily. The instructions of HLL are written in English statements. The programs of HLL are not directly executed on the computer. A language translator is required to translate the HLL into low level language.

Q.7 What is low-level language?

Ans. The language which is close to machine language is called low level language. A Computer can easily understand low level language. The low level language requires a deep understanding of the machine architecture.

Q.8 What is machine language?

Ans. It is the native language of computer. Every machine language instruction consists of 0's and 1's. It is difficult for human beings to understand and learn it. And it is also difficult to locate and remove errors in the program. It is directly executed by the computer. The machine language programs are machine dependent.

Q.9 What is assembly language?

Ans. Machine language instructions(0's and 1's) are replaced with English like words known as mnemonics (Ne-monics) It is also called symbolic language. An assembler is required to translate the assembly language programs into machine language

Q.10 What is meant by portability?

Ans. The programs written in High level language are closed to human language and programs are machine independent i.e. a program can be run on different types of computers.

Q.11 Define source code.

Ans. The program written in High Level Language is called source program. The computer does not understand the source code. The source code is converted into machine code and then it is directly executed on the computer.

Q.12 What is Object Program?

Ans. The program in a machine language is called object program. The computer understands the object code directly.

Q.13 What is Linker?

Ans. The linker is a program that combines the object program with additional library files and produces one executable file with .exe extension. Linking is a process in which the object file is produced by the compiler is linked to many other files by the linker.

Q.14 What is Loader?

Ans. For execution, the loader loads the executable files in the memory. It is also system software. Ctrl+F9 key is used to load and run the programs. The executable file (.exe) runs directly on the computer after loading process.

Q.15 What is a Language Processor/Translator?

Ans. It is software that is used to translate the high-level language programs into machine language. Each language has its own translator. Only one type of translator is used in any language. There are three types of language processor.

Compiler, Interpreter and Assembler

Q.16 What is a Compiler?

Ans. The language translator translates the source code into object code and the whole program is translated at the same time. If a program contains errors then compiler cannot convert the source code into machine code until all the errors are removed from the source program.

Q.17 What is an Interpreter?

Ans. The language translator translates the source code into object code statement by statement. The working of interpreter is slower than the compiler. Any error in the program stops execution and all the instructions are executed in a program before errors in the program.

Q.18 What is an Assembler?

Ans. The language translator translates the assembly language into machine code.

Q.19 What are PROGRAMMING ERRORS?

Ans. The errors in a program are called BUGS. The process of finding and removing these errors is called debugging. There are three types of errors.

Syntax Errors

Run Time Errors

Logical Errors.

Q.20 What are Syntax Errors?

Ans. A syntax error occurs when the program violates one or more grammar rules of high-level language. A compiler detects these errors at the time of compilation. The errors must be removed for the successful compilation. These errors are easy to locate and remove because the compiler specifies the location and type of error.

Q.21 What are Run Time Errors?

Ans. A run time error occurs when the program directs the computer to perform an illegal operation. Run time errors are detected and displayed by the computer during the execution of a program. When a run time error occurs, the computer stops the execution and displays a diagnostic message. Examples: Dividing number by zero.

Q.22 What are Logical Errors?

Ans. A logical error occurs when a program follows a faulty algorithm. The compiler cannot detect logical errors; therefore no error message is reported from the compiler. Programs cannot be crashed due to logical errors. It is difficult to detect logical error. Program with logical errors produces wrong output.

Q.23 Define Preprocessor Directives.

Ans. The instructions given to the compiler before the beginning of the actual program are called preprocessor directives. It always begins with # symbol e.g. #include, #define.

Q.24 What is #include directive:

Ans. It is a preprocessor directive. The include directive gives a program access to the library. The include directive tells the compiler where to find the meaning of identifiers used in the program.

Q.25 What is #define directive or constant Macro?

Ans. This preprocessor directive is used to define a constant macro. Constant Macro is a name that is replaced by a particular constant value before compilation. It cannot be changed during program execution.

Syntax:

#define Macro_name expression/constant

Example: #define PI 3.142857

Q.26 What is a Preprocessor?

Ans. Preprocessor is a program that modifies or handles a C program prior to its compilation.

Q.27 What are Header Files?

Ans. These files contain the definition of standard library functions. The extension of header file is .h Each header file contains definition of one type of functions only. The include directive is used to add header files in the program. All the header files are located in INCLUDE subfolder.

#include <header file name>

#include "header file name"

Q.28 What is a Main Function?

Ans. It indicates the beginning of a C program. Every C program has a main function and if it is not included in the program, then the compiler generates an error message.

Syntax:

```
void main (void)
```

```
{
```

```
Statements of the program // Body of main function
```

```
}
```

Q.29 What are Delimiters?

Ans. The braces in the main function represents start and end of the program, these braces are called delimiters.

```
{      represents start of the code.
```

```
}      represents end of the function code.
```

Q.30 What is a Statement Terminator?

Ans. Each statement of a C program ends with a semicolon (;) called statement terminator. If statement terminator (semicolon) is missing in any statement then the compiler will generate the following error message:

Statement missing;

Q.31 What is Structured Programming Language?

Ans. A programming language in which the logic of the program is divided into a number of smaller sections. Each section of the program performs a specific function. These programs are easy to write, debug and modify.

Q.32 What is Un-Structured Programming Language?

Ans. A programming language in which the logic of the program is written in a single module. It is very difficult to detect any error in the program. Its readability is difficult.

Q.33 List two reasons why it would be preferable to write a program in C rather than machine language.

Ans. Each instruction in machine language consists of 0's and 1's, therefore it is difficult to understand and learn it. Moreover it is difficult to locate and remove errors in the program.

The programs written in C language are closed to human language and programs are machine independent i.e. a program can be run on different types of computers.