CHAPTER 2

PROGRAMMING IN C

SHORT AND LONG QUESTIONS

Q1. Define computer programming.

Ans: Computer Programming:

Computer Programming is the process of writing a computer program in computer language to solve a particular problem.

Computer program controls the operation of a computer and it is developed in a computer language to perform a task. Computer languages are also known as programming languages.

Q2. Define Programming Language. Explain low level and high-level languages.

OR

Explain popular programming languages and their uses.

Ans: Programming Language:

A programming language is a language which is understood by computer. It is designed to give instructions to a computer to perform a specific task. It is used to write computer programs.

Classification of Programming Languages:

Programming languages can be classified into two categories that is low level language and high-level language.

i. **Low Level Languages:**

Low level language is machine-oriented language. To understand low level language, detailed knowledge of internal working of computer is

required. Low level languages include machine language and assembly language.

Classification of Low Level Languages:

- **Machine Language**

Programming language that is directly understood by computer hardware is known as machine language. Machine language is associated with architecture of computer. Therefore, programs written in machine languages for one computer will not work on another because of design differences. It consists of zeroes and ones. It is almost impossible for humans to use machine language because it entirely consists of numbers. Therefore, practically no programming is done in machine language instead, assembly language and high-level languages are used.

- **Assembly Language:**

Assembly language consists of symbolic codes or abbreviations known as mnemonics. It was developed to make computer programming easier than machine language. The abbreviations used in assembly language make it easier to learn and write programs compared to machine language. A program written in assembly language must be converted into machine language before it is executed by computer. A program known as assembler is used to translate assembly language into machine language.

Characteristics of Assembly Language:

Some important characteristics of Assembly language are

- Assembly language allows programmers to have access to all the special features of the computer they are using. Certain types of operations which are not possible in high-level languages are easily programmed using assembly language.

- Generally, a program written in assembly language will require less storage and less running time than one prepared in a high-level language.
- Assembly languages are still the best choice in some applications but their use is gradually declining.

ii. **High-level Languages (HLLs):**

High-level languages are English-oriented languages and they are commonly used for writing computer programs. These languages use English language word such as print, goto, if, end etc. Therefore, they are easy to learn and use.

Examples: Some examples of high-level languages are Visual Basic, C, Java and Pascal.

A program known as complier/interpreter is required to translate a high-level program into machine language. Coding and debugging of a high-level language program are much easier than a program written in a low level language.

Classification of High-level Languages:

High-Level languages can be classified into procedural, structured and object-oriented programming languages.

- **Procedural Languages:**

Procedural programming is based upon the concept of modular programming. In modular programming, programs are divided into smaller parts known as modules. Modular programs consist of one or more modules. A module is a group of statements that can be executed more than once in a program. Each module in a program performs a specific task.

Advantages of procedural languages:

It is easy to design, modify and debug a program in a procedural language since it provides better programming facilities.

Example of procedural languages:

Some examples of procedural languages are FORTRAN, Pascal, C and BASIC.

- **Structured Languages:**

  Structured languages consist of three fundamental elements, which are sequence, selection and repetition.

Sequence:

It means writing program statements in a logical sequence. Each step in the sequence must logically progress to the next without producing any undesirable effects.

Selection:

It allows the selection of any number of statements based on the result of evaluation of a condition which may be true or false. Examples of statements that implement selection in programming are if, else-if, switch etc.

Repetition (loop)

It means executing one or more statements a number of times until a condition is satisfied. Repetition is implemented in programs using statements, such as for and while loops.

Some examples of structured languages are ALGOL, PL/1, Ada and Pascal.

- **Object-Oriented Programming Languages (oops):**

Object-orient programming (oops) refers to a programming method that is based on objects such as student, vehicle, building etc. Object-oriented programming language provides a set of rules for defining and managing

objects. An object can be considered a thing that can perform a set of activities.

For example, the object vehicle can be defined as an object that has number of wheels, number of doors, color, number of seats etc. The set of activities that can be performed on this object include Steer, Accelerate, Brake etc.

The most widely used object-oriented programming language are C++, C#, php and Java.

Q3. What is programming environment?

OR

Describe the concept of integrated development environment.

Ans: Programming Environment:

Programming environment is the set of process and programming tools used to develop computer programs. In the past, programmers used various standalone programming tools for developing computer programs. These included editors, complier, linker, debugger etc. Using separate programs provided a difficult and time-consuming environment for creating computer programs.

Today, programmers use Integrated Development Environment for developing computer programs. Integrated Development Environment is an application package that consists of editor, compiler, linker and debugger with a graphical user interface.

Q4. Describe different integrated development environment modules of C language.

OR

Define the following terms of C languages (Editor, Compiler, Linker, Loader and Debugger)

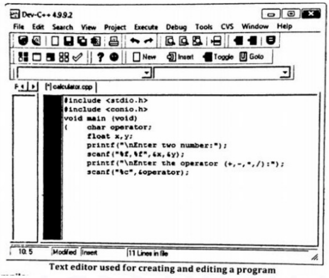
Ans: Programming Environment of C language:

A C language IDE consists of the following modules

- Text Editor
- Compiler
- Linker
- Loader
- Debugger

Text Editor:

A text editor is a simple word processor that is used to create and edit source code of a program as shown in Fig. Files created by a text editor are plain text files. Most of the editors automatically highlight compile errors to simplify removing them.

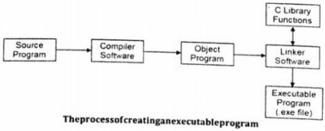


```
#include <stdio.h>
#include <conio.h>
void main (void)
{     char operator;
      float x,y;
      printf("\nEnter two number:");
      scanf("%f,%f",&x,&y);
      printf("\nEnter the operator (+,-,*,/):");
      scanf("%c",&operator);
```

Text editor used for creating and editing a program

Compiler

A compiler is a computer software that translated C language program (source program) into machine code (object program) that can be understood and executed by the computer as shown in Fig. It also detects syntax errors in the source program and gives hints to the programmer to correct them. A compiler will only create object program if all the syntax errors in a program have been corrected if they existed. A program written in C language has the extension .c (for example first c) and after compilation the object program extension will be .obj (for example first.obj)



Theprocessofcreatinganexecutableprogram

Linker

Linker is a software that translates object program into a single executable program as shown in Fig. During this process, if it could find the definition of a particular function that is used in the program, then it would assume that it is defined in C library. It will replace this function in the object with the code from C library and then create a single executable program.

Loader:

It is a software that loads programs into memory and then executes them.

Debugger:

It is a software that executes a program line by line, examines the values stored in variables and helps in finding and removing errors in programs.

Q5. List the uses of C language.

Ans: uses of C language:

C is a popular and widely used programming language for creating computer programs. A large variety of application programs and many modern operating system are written in C.

Q6. Explain C language character set.

Ans: C Language Character Set:

The C Language character set includes:

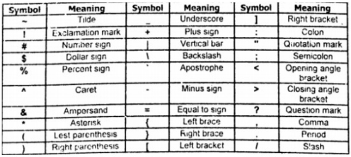
Letters:

C language comprises the following set of letters to form a standard program. They are

- A to Z in Capital letters
- A to z in small letters
- In C programming, small letter and caps letter are distinct.

Digits:

- C language contains the following special character in association with the letters and digits



Symbol	Meaning	Symbol	Meaning	Symbol	Meaning
~	Tilde	_	Underscore	]	Right bracket
!	Exclamation mark	+	Plus sign	:	Colon
#	Number sign	|	Vertical bar	"	Quotation mark
$	Dollar sign	\	Backslash	;	Semicolon
%	Percent sign	'	Apostrophe	<	Opening angle bracket
^	Caret	-	Minus sign	>	Closing angle bracket
&	Ampersand	=	Equal to sign	?	Question mark
*	Asterisk	{	Left brace	,	Comma
(	Lest parenthesis	}	Right brace	.	Period
)	Right parenthesis	[	Left bracket	/	Slash

Q7. Define header files.

Ans: Header Files:

C language contains a number of standard functions in library file that performs various tasks in C programs. These tasks include all the input/output

operation and all the math operations. Library file contains header files and each header file contains a set of functions.

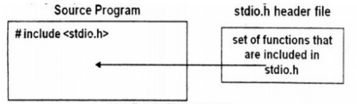
Q8. Write the uses of header files.

Ans: Uses of Header Files:

Some commonly used header files are stdio.h, conio.h and math.h

- Some commonly used functions that are included in stdio.h file and printf(), scanf() etc.
- Some of the functions that are included in conio.h file are clrscr(), getch() and getche().
- The math.h header file contains mathematical function such as sqrt(), pow(), log(), sin(), cos() and tan().

In order to use a library function, the programmer must include appropriate header file at the beginning of the program as shown in figure. When source program is translated into executable file, a copy of code of function is pasted in the source program by the linker from the header file before creating the executable file.



UseofheaderfileinaCprogram

Q9. What is the structure of a C program?

Ans: Structure of a C Program:

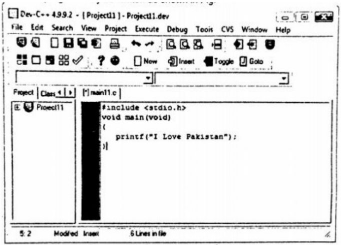
The formal according to which a program is written is called the structure of program. The following I the structure of C program.

Preprocessor directives
Global declarations
Main function
{
Local declaration
Statements } Body of main() function
}
User-defined function
Function1
Function 2
Function 3 } (option to user)

To introduce the structure of a C program, consider a very simple program that will print the message.

I love Pakistan:

The program may be written as shown in Fig.


Dev-C++ 4.9.9.2 - [Project11] - Project11.dev
File Edit Search View Project Execute Debug Tools CVS Window Help
New Insert Toggle Goto
Project Classes
[*] main11.c
Project11
5: 2 Modified Insert 6 Lines in file

```
#include <stdio.h>
void main(void)
{
    printf("I Love Pakistan");
}
```

Programtoprintamessageonthescreen

This program consists of three main parts which are preprocessor directive, the main() function and the body main().

Q10. Explain the types of comments in C language:

Ans: Types of Comments:

There are two types of comments.

i. Single line comment
ii. Multiple line comment

i. Single Line Comment

The // is used as single line comment. The syntax of single comment is

// comment

An example is:

// Programmer: M. Sajjad Header

ii. Multiple Line Comment:

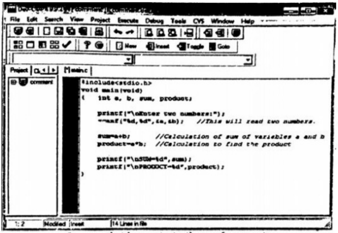
The /*   */ is used for multiple line comments.

Comment can span to multiple lines as shown in the following example

/* Programmer M. Sajjad Header

Programming Language C

Date Program was Written: 15-04-2014 */



```
#include<stdio.h>
void main(void)
{   int a, b, sum, product;

    printf("\nEnter two numbers:");
    scanf("%d,%d",&a,&b);    //This will read two numbers.

    sum=a+b;        //Calculation of sum of variables a and b
    product=a*b;    //Calculation to find the product

    printf("\nSUM=%d",sum);
    printf("\nPRODUCT=%d",product);
}
```

Program that demonstrates the use of comment

The program in figure demonstrates how single line comments are used in a program. In this program comments describe the purpose of statements.

Q11. Explain the different data types used in C programs.

Ans: Data Types Used in C Programs:

C provides three main data types for variables, that is, integer, floating-point and character variable.

- **Integer Data Type:**

  Integer variable declaration statement has the form:

Type specifier Variable;

For example: **int sum;**

The declaration consists of the type name, int, followed by the name of the variable, sum. All the variables used in a C program must be declared. If there are more than one variable of the same type, separate the variable names with commas as shown in the following declaration statement.

Declaring a variable tells the computer the name of the variable and its type. This enables the compiler to recognize the valid variable names in program. This is very helpful if the user types the wrong spellings of a variable name in his program, the compiler will give an error message indicating that the variable is not declared.

Declaration of an integer variable can begin with the type qualifiers, short, long, unsigned o signed.

Some examples are: **short int num;**

Long int sum, avg;

Unsigned int count;

Short unsigned int count;

Type qualifiers refers to the number of bytes used for storing the integer on a particular computer. Refer to computer manual to find out the sizes of the computer being used. Generally, the number of bytes set aside by the compiler for the above type qualifiers are as given in table.

Integer data types used in C:

Variable Declaration	No. of Bytes	Range
--	2	-32,768 ~ +32,767
Long int	4	-2,147,483,648 ~ +2,147,483,647
Unsigned int	2	0 ~ 65,535

Here unsigned implies that the value of variable is greater than or equal to zero. The qualifier, signed is rarely used since by default, an int declaration assumes a signed number. Note that in the above examples, the word int may be omitted when it begins with long or unsigned type qualifiers in the declaration statements.

For example, you can write: **short num;**

- **Floating-point Data Type:**

Floating-point variables are used for storing floating-point numbers. Floating point numbers are stored in memory in two parts. The first part is the **mantissa** and the second part is the **exponent.**

The **mantissa** is the value of the number and the **exponent** is the power to which it is raised. Some examples of floating-point variable declaration statements are:

Float base, height;

Double float a, b, total;

Long double float size, x,y;

Usually the number of bytes set aside by the compiler for the above floating-point type qualifiers are as given in table.

Floating-point data types used in C:

Variable Declaration	No. of Bytes	Range
Float	4	10^{-38} ~ 10^{38} (with 6 or 7 digits of precision)
Double float	8	10^{-308} ~ 10^{308} (with 15 digits of precision)
Long double float	10	10^{-4932} ~ 10^{4932} (with twice the precision of double)

Here, precision means the number of digits after the decimal point. The word float may be omitted when it is preceded by double or long double type qualifiers in the declaration statements.

There is another method of representing floating-point numbers known as exponential notation.

Example:

For example, the number 23,688 would be represented as 2,3688e4. Here 23688 is the value of the number (mantissa) and 4 is the exponent. The exponent can also be negative.

Example:

For example, the number 0.0005672 is represented as 5672 in exponential notation. Exponential notation is used to represent very large and very small numbers. In C, a floating point number of type float is stored in four bytes. Three

bytes for the mantissa and one byte for the exponent and provides 6 or 7 digits of precision.

- **Character Data Type:**

Character variables are used to store character constants. Examples of declaration of character variable are:

Char ch, letter, a;

A character variable can only store one character. The compiler sets aside one byte of memory for storing a character in a character variable.

Q12. Define constant qualifier.

Ans: The Constant Qualifier:

The constant qualifier is used with a data type declaration to specify a constant value.

Examples: Some examples of const qualifiers are

Const int COUNT= 5;

Const float PI=3.14;

The same thing can also be achieved using #define preprocessor as given below

#define COUNT 5

#define PI 3.14

In most of the situations there is no difference between using define or const qualifier. The only difference is defined is a preprocessor directive that simply replaces a symbol with a constant value throughout the program. The const qualifier declares a variable type and assign it a constant value. It has some advantage at advance level of programming in C language.

Q13. Explain the main () Function.

Ans: The main () Function:

A C program consists of one or more functions. C uses functions as the building block of its programs. A function performs a single well-defined task. Every C program must have the function main () which is the first section to be executed when the program runs.

The word void before the function main () means that this function does not return a value and the second void inside the brackets means it does not have any argument.

A program may also contain user-defined functions. User-defined functions are written by programmers and they are included in the library file. These functions may be written before or after the main () function and are used inside the main () function. When we run a program, the main () function is always executed first no matter where it appears in the program.

The Body of main () Function:

The body of the function main () is surrounded by braces (curly brackets { and }). The left brace indicates the start of the body of the function and the matching right brace indicates the end of the body of the function.

The body of the function in the program (see Figure) consists of a single statement printf() which ends with a semicolon(;) printf() is the standard output function. The text to be printed is enclosed in double quotes. A statement in C is terminated with a semicolon. Therefore, a semicolon is used at the end of **printf()** statement. This program will print the message **I Love Pakistan** on the screen.

Q14. Differentiate between Global and Local Variables.

Ans: Global and Local Variables:

Variables are used in C programs to store values. All the variables used in a program must be declared. There are two types of variables, global variables and local variables.

In the program structure, global variables are declared before the main() function and local variables are declared within it.

Global variables are those variables which can be used in any function in the program whereas local variables are used only within a function in which they are declared.

KEY POINTS

- **Syntax** refers to the rules of a programming language according to which statements of a program are to be written.
- **Semantic** gives meaning to statements of a programming language.
- **Machine language** is a programming language that is directly understood by computer hardware. It consists of zeroes and ones.
- **Assembly language** consists of symbolic codes or abbreviations. It was developed to make computer programming easier than machine language. A program called assembler is required to convert assembly language to machine language for execution by the computer
- **High-level language** is an English-oriented language. It is commonly used for writing computer programs. A program called compiler/interpreter is required to translate high-level language into machine language for execution by the computer.
- **Compiler** is a computer program that translates a program written in a high-level language into machine language equivalent program. It converts the entire program into machine language before execution by the computer.

- **Interpreter** is a computer program that translates a program written in a high-level language into machine language. It translates one instruction at a time and executes it immediately before translating the next instruction.
- **Programming environment** is the set of processes and programming tools used to develop computer programs.
- **Integrated Development Environment (IDE)** is a computer software that is used to create, compile and run programs. It brings all the processes and tools required for program development into one place.
- **Text editor** Is a simple word-processor that is used to create and edit source code of a program.
- **Linker** is a software that takes one or more object files, generated by a compiler and combines them into a single executable program.
- **Loader** is a software that loads programs into memory and then executes them.
- **Debugger** is a software that executes a program line by line, examine the values stored in variables and helps in finding and removing errors in programs.
- **Header file** is a file that contains predefined functions of C language. Including a header file in a C source produces the same result as copying the header file into the source file.
- **Reserved words** are those words that are part of a programming language and have special purpose in computer programs.
- **Preprocessor directives** are instructions for the C compiler at the beginning of program.
- **Comments** are notes that are included in programs when a fact is necessary to be brought to the attention of program's reader.
- **Constant** is a quantity whose value does not change during program execution.

- **Variable** is a symbolic name that represents a value that can change during execution of a program.
- **Type casting** is a way to convert a variable from one data type to another data type.

EXERCISE

Q1. Select the best answer for the following MCQs.

i. **What defines the rules of valid statements in programming?**

A. Compiler
B. Interpreter
C. Syntax
D. Semantic

ii. **Which language is directly understood by the computer?**

A. Machine language
B. Assembly language
C. High-level language
D. C language

iii. **When was C language developed?**

A. Late 1960s
B. Early 1970s
C. 1980s
D. 1990s

iv. **Who developed Java language?**

A. Dennis Ritchie
B. Microsoft
C. Sun Microsystems
D. IBM

v. **What is the other word used for Reserved Words?**

A. Compiler words
B. Keywords
C. Special programming words
D. Mnemonics

vi. **How many bytes are set aside by the compiler for a variable of type int?**

A. 2
B. 3
C. 4
D. 5

vii. **How many bytes are set aside by the compiler for a variable of type float?**

A. 2
B. 3
C. 4
D. 5

viii. **What is the range of numbers that can be stored in a variable of type double float?**

A. -32,768 ~ +32,767

B. $10^{-38} \sim 10^{38}$

C. $10^{-308} \sim 10^{308}$

D. $10^{-4932} \sim 10^{4932}$

ix. Which program translates high-level language into machine language?

A. Compiler

B. Linker

C. Loader

D. Debugger

x. Which software helps in finding and removing errors in programs?

A. Linker

B. Text Editor

C. Loader

D. Debugger

Answers

i.	C	ii.	A	iii.	B	iv.	C	v.	B
vi.	A	vii.	C	viii.	C	ix.	A	x.	D

SHORT QUESTIONS

Q2. Give short answers to the following questions:

i. Define computer program.

Ans: Computer Program:

A computer program is a set of instructions (statements) written in a programming language to solve a particular problem and achieve specific results. Any task performed by a computer is controlled by a set of instructions that are executed by the microprocessor. A large variety of programming languages have been developed for writing computer

programs to use the computer as a problem-solving tool. Each statement of a programming language has syntax and semantic.

ii. Differentiate between syntax and semantic.

Ans: Syntax:

Syntax refers to the rules of a programming language according to which statements of a program are to be written. It describes the way to write correct statements in a program. Syntax of a programming language is similar to the grammar of a natural language.

For example, an assignment statement consists of a variable and an expression separated by equal sign as an assignment operator. This is the syntax of assignment statement and it can be expressed as given below

Variable=expression

Semantic:

Semantic gives meaning to statements of a programming language. It describes the sequence of operations to be performed by a computer when executing the statements of a computer program.

For example, in the assignment statement

Sum = a + b ;

The semantic of the statement is to perform the expression, that is, add the values stored in variables a and b and then store the result in variable sum.

iii. Write three differences between assembly language and HLLs.

Ans: Difference between high-level language and assembly language:

Features of high-level language (HLLs):

- **i.** High-level languages are easily understandable.
- **ii.** The programs that are developed in high-level languages are portable.
- **iii.** In case of high-level languages debugging of the code is easy and the program written is not machine dependent.

Features of Assembly language:

- **i.** Although Assembly level language are not easy to understand they are relatively easier as compared to machine level language.
- **ii.** The programs written in this language are not portable and the debugging process is also not very easy.
- **iii.** The programs developed in assembly language are thoroughly machine dependent.

iv. Write four characteristics of HLLs.

Answer: Characteristics of High-level Languages:

High-level languages have the following characteristics:

1) These languages were developed to make computer programming simple, easier and less prone to errors.
2) High-level languages are not machine dependent. They enable programmers to write programs that are independent of a particular type of computer.
3) Programs written in high-level languages must be translated into machine language by a compiler or an interpreter before execution by the computer.
4) The process of finding and removing errors in programs (debugging) is easier in high-level languages compared to low level language.
5) High-level language programs are highly structured. They allow programmers to break lengthy programs into a number of modules which

can be written and tested independently. This makes writing and testing of programs easier.

v. Define Integrated Development Environment (IDE).

Ans: Integrated Development Environment (IDE):

IDE is computer software that brings all the processes and tools required for program development into one place. IDE's aim is to make the life of programmers easier by grouping together all the tasks needed for building applications into one environment.

Today's modern IDEs have user-friendly Graphical User Interface (GUI). Most of the new programming languages use Integrated Development Environment (IDE) to create, compile and run programs.

vi. Differentiate between constant and variable.

Ans: Constants and Variables:

Constants and variables are used in expressions in computer programs. After an expression is evaluated, the result of the expression is also stored in a variable.

- **Constant:**

  Constants are quantities whose values do not change during program execution.

Types of Constant:

They may be numeric, character or string.

i. Numeric Constants:

Numeric constants are of two types, integer and floating-point numbers.

Integer constants represents values that are counted like the number of students in a class. Some examples of integer constants are 7145, -234, 26 etc.

Floating point constants are used to represented values that are measured like the height of a person which might have a value of 166,75 cm or the weight such as 82.6 kilograms.

ii. Character Constant

Character Constant is one of the symbols in C character set. It includes digits 0 to 9, upper case letters A to Z, lowercase letters a to z, punctuation symbols such as semicolon (;), comma (,), period(.) and special symbols such as +, -, =, >, etc. A character constant is enclosed by single quotes such as 'a', 's' etc.

iii. String Constant:

String Constant contains a string of characters within double quotes such as "Hello Ahmed", "a" etc.

- **Variable:**

  A variable is a symbolic name that represents a value that can change during execution of a program. A variable has a name, known as variable name and it holds data of other types. A number or any other type of data held in a variable is called its value. It also indicates the types of value a variable can represent.

Type of Variables:

Variables are of two types, numeric and character.

i. Numeric Variables:

Numeric variables are used to represent numeric values in computer programs. They represent integer and floating-point values. Some examples of numeric variables are sum, length, salary, marks etc.

ii. Character variables:

Character variables represent character values in computer programs. It can represent a single character or a string of characters. Some examples of character variables are name, city, gender etc.

When a variable is used in a computer program, the computer associates it with a particular memory location. The value of a variable at any time is the value stored in the associated memory location at that time. Variables are used so that the same space in memory can hold different values at different times.

vii. Which of the following are valid C variables? Give the reason if not a valid variable area, 5x, Sum, net pay, float, _age, else, case, size22, my_weight

Ans: Valid:

- area
- Sum
- _age
- size22
- my_weight

Not valid:

5x	A variable name cannot start with a number
net pay	Space cannot be used in variable name
float	Data types cannot be used as a variable name
else	Commands or reserved words cannot be used as a variable name
case	Commands or reserved words cannot be used as a variable name

viii. What are reserved words? Why they should not be used as variable names?

Ans: Reserved Words:

The words that are part of programming language and have special purposes in computer programs are called reserved words or keywords.

Reserved words have predefined use and cannot be used for any other purpose. Reserved words are always in lowercase. There are 32 words defined as reserved words in C. A complete list of reserved words used in C language is given in table below.

Reserved words cannot be used as variable names since the compiler needs to distinguish between them in order for the code to work properly. Using reserved words as variable name might cause unintentional results, that's why it isn't allowed in C language.

List of reserved words of C:

Auto	Double	Int	Struct
Break	Else	Long	Switch
Case	Enum	Register	Typedef
Char	Extern	Return	Union
Const	Float	Short	Unsigned
Continue	For	Signed	Void
Default	Goto	Sizeof	Volatile
Do	If	Static	while

ix. Why comments are used in programs?

Ans: Comments in C Language:

It is a good programming practice to add comments in program to make it easy for others to understand it. Comments in the source code are ignored by the compiler.

Comments are added in programs when a fact is necessary to be brought to the attention of program's reader. Generally, programmers write comments at the beginning of the program explaining the reader what the program is intended to achieve and inside the program code where something is to be clarified about the structure of the program.

There are two types of comments

- Single line comment
- Multiple line comment

x. What is the purpose of header files in C language?

Ans: Purpose of Header File in C Language:

C language has many libraries and predefined functions which make programming much easier. In order to use these libraries and functions, we include the header file at the beginning of the program by using the preprocessor directive. Header files contain a number of predefined standard functions that carry out tasks including all the input/output and math operations. Different header files contain different functions.

Some commonly used header files are stdio.h, conio.h and math.h.

EXTENSIVE QUESTIONS

Q3. Describe the following High Level languages.

- **a) C/C++**
- **b) Visual Basic**
- **c) C#**
- **d) Java**

Ans: Popular High Level Languages (HLLs):

a) C/C++

C language was developed in early 1970s by Dennis Ritichie at Bell Laboratories. C has become one of the most popular programming languages today. It is a highly structures programming language that is easy to understand and use. In the past, it was mainly used for writing system programs such as operating system, compilers, assemblers, etc.

Today, it is used for writing all type of application programs as well, such as word-processing programs, spreadsheet programs, database management system, educational programs, games etc.

C++ was developed by Bjarne Stroustrup also at Bell Laboratories during 1983-1985. C++ is a superset of C, meaning that any valid C program is also a valid C++ program. The purpose of developing C++ was to provide programming facilities to easily and quickly write more powerful programs.

b) Visual Basic:

Visual Basic (VB) is a high-level language which evolved from the earlier version called BASIC. BASIC stands for Beginner's All-purpose Symbolic

Instruction Code. VB is a very popular programming language for writing Windows and Web applications. It provides a graphical development environment to programmers to develop powerful Windows and Web applications.

VB is commonly used for developing business programs such as payroll system and inventory control program. The user can also write programs related with engineering, science, arts, educations, games etc.

c) C#:

C# is a language developed in 2000 by Microsoft Corporation. It is a simple, modern, general purpose programming language.

Syntax of C# is very similar to C and C++. It also has some features of Java. It is a language that makes computer programming easy and efficient. It provides facilities to write Web applications that can be used across the Internet.

All types of programs including games, utilities, operating systems, compilers, business applications and Web based applications can be developed in C#.

d) Java:

Java is a high-level language developed by Sun Microsystems. It is very similar in syntax to C and C++. In Java, the user can write all types of programs as those written in other programming languages and small programs that can be embedded in a Web page accessed through Internet. Java is an ideal language for network computing. It is used for writing programs for a wide range of devices, computers and networks. It is widely used in Web Application.

The current versions of most of the Web browsers are made Java enabled. A few browsers that support Java and Microsoft's Internet Explorer, Firefox and Mozilla.

Q4. What is C language IDE? Explain it modules in detail.

Ans: Integrated Development Environment (IDE)

Programming environment is the set of processes and programming tools used to develop computer programs. Previously programmers had to use many separate programming tools including editor, compiler, linker, debugger etc. to develop computer programs. Using separate programs was difficult and time consuming for programmers. Nowadays Integrated Development Environment is used by programmers making it easy for them to code and make programs.

An **Integrated Development Environment (IDE)** is a computer software that brings all the processes and tools required for programming into one place. IDEs combine all the tools needed for building applications into a single environment. Modern IDEs have user-friendly Graphical User Interface (GUI) making programming easier and more intuitive.

Programming Environment of C Language:

A C language IDE comprises separate various modules. A subdivision or a block of the overall environment is known as a module. A C language IDE is made up of the following modules.

- **Text Editor**

A text editor is a simple word-processor that is used to create and edit source code of a program. Files created by text editor are plain text files. Most of the editors automatically highlight compilation errors so that they can be easily removed.

- **Compiler**

A compiler is a computer program that translates C language (source program) into machine code (object program) that can be understood and executed by the computer. It also detects syntax errors in the source program

and provides various hints on how to correct them. Compiler only converts the source program into object program is it is free of any syntax error. Program written in C language has the **.c** extension and after compilation the object program extension will be **.obj**.

- **Linker**

Linker is the software that translates the object program into a single executable program. If linker cannot find the definition of a particular function used in the program, then it assumes that it is defined in the C library. This function in the object program will be replaced with the code from the C library and then a single executable file with the extension **.exe** will be created

- **Loader**

Loader is the software that loads a program into the computer memory and then executes it.

- **Debugger**

Debugger executes the program line by line, examines the values stored in all the variables and helps in finding the errors in the program. These errors can then be removed by the programmer.

Q6. What are the rules for specifying a variable name in C language?

Ans: Rules for Specifying Variable Names in C language:

A variable is a symbolic name that represents a value that can change during the execution of a program. A variable has a name and it holds data of a specific type. The data that is held in the variable is known as its value.

The following are the rules for specifying variable names in C language.

- A variable begins with a letter or underscore (_) and may consist of letters, underscores and/or digits
- The underscore may be used to improve readability of the variable name

For example, over_time

- There is no restriction on the length of a variable name. However only the first 31 characters of a variable are significant. This means that if two variables have the same first 31 characters, they are considered to be the same variables.
- Both upper-case and lower-case letters are allowed in naming variables. An upper-case letter is considered different from a lower-case letter. For example, the variable AVG is different from Avg or avg.
- Special characters cannot be used as variable name, e.g #, ?, @ etc
- Reserved words of C languages such as **int**, **case**, **if** etc., cannot be used as variable names.
- Space is not allowed and there must be no embedded blank in the name of a variable. For example, **ma ss** is not a correct variable name.

Q8. What is a preprocessor directive? Explain #include preprocessor directive in detail.

Ans: Preprocessor Directives:

Preprocessor directives are instructions for the C compiler. Every C language program contains certain preprocessor directives at the beginning of the program. Before translating a C language program into machine language, the compiler of C language carries out the processor directives. These directives start with number sign (#)

The most commonly used preprocessor directives are **#include** and **#define**

- **#include Preprocessor Directive:**

  It has the following syntax

  #include<header file name>

When the preprocessor is carried out by the C compiler, it will search for the header file that is written within the less than (<) and greater than (>) symbols and copy it into the source file.

For example, if the **stdio.h** header file is written within the less than and greater than symbols then the **#inlcude** preprocessor directive will tells the C compiler to copy the **stdio.h** header file into the program. The **stdio.h** header file stands for standard input-output header. It includes the standard **printf()** and **scanf()** function prototypes. After including the header file, we can then use the standard library functions that are included in it.