14-10-21

Subject Code : 01CT0101

Subject Name : Introduction to Computer Programming

Shashank N. Bagda
92100133020

* Question - Answer :

1) Why to learn C language ?

→ At present there are so many languages in the world. So basically a language is a thing which helps a programmer & the computer to communicate with each other. Some of the languages are used to develop application & software while some are used to built new software & application.

- C is a core language :

In computing, C is a general purpose, cross-platform, block structured procedural, imperative computer programming language. There are so many languages based on C. If someone learn C, it will be much easier to learn languages. Such languages include C++, Java & Perl.

- C is a small language :

- C has only thirty - two keyboards & only twenty of them are common use. This makes it relatively easy to learn compared to bulkier languages.

- C is quick :

- We can write codes which run quickly, & the program can be very close to the hardware. By that, you can access low level facilities in your computer quite easily, without the compiler or run time system stopping you from doing something potentially dangerous.

- C is portable :

- C programs written on one system can be run with little or no modification on other systems. If modification are necessary, they can often be made by simply changing a few entries in a header file accompanying the main program. The use of compiler directives to the pre-processor makes it possible to produce a single version of a program which can be compiled on several different types of computer.

In this sense C is said to be very portable. The function libraries are standard for all versions of C so they can be used on all systems.

2) List Escape Sequence available in C.

→ \a	→	Ring terminal bell
\?	→	Question Mark
\b	→	Backspace
\r	→	Carriage return
\f	→	Form Feed
\t	→	Horizontal tab
\v	→	Vertical tab
\0	→	ASCII null character
\\	→	Back slash
\"	→	Double quote
\'	→	Single quote
\n	→	New line
\o	→	Octal constant
\x	→	Hexadecimal constant

3) List C keywords.

→ auto
break
case
char
const
continue

default
do
double
else
enum
extem

Float
for
goto
if
int
long

register
restrict
return
short
signed
sizeof

static
stuct
switch
typedef
union
unsigned

void volatile while Bod Complex Imaginary

4) List the rules providing variable name to C.

- A variable names are combination of alphabets or digits or underskore.
- The first character is always a Letter.
- Length of variable name can range from 1 to 8
- A space in between is not allowed
- If we want to seperate name then we can use underscore instead of space.
- Special Characters like commas, are allowed in variable name.
- In case of C language it a case sensitive language which means a variable name declared as ~~fla~~ Flag is not same as FLAG. They both will be treated as different variable.
- There are certain reserved words in C language, known as keywords. Words similar to a keyword can not be used as a variable name.

5) List C Tokens

→ Tokens are the basic lexical building blocks of source code. In other words, one or more symbols understood by the compiler that help it interpret your code.

- Characters are combined into tokens according to the rules of the programming language.
- The compiler checks that the tokens can be formed into legal strings according to the syntax of the language.
- There are five classes of tokens: identifiers, reserved words, operators, separators & constants.
- The tokens that will be generated

Keywords	:	if, else	• Bold Black
Identifier	:	x (variable) (name)	Black
Constants	:	2, 10, 5, 'A'	
Operators	:	+, =, ++, sizeof	
Separator	:	;	
String	:		• Blue colour
~~Special Symbol~~	~~:~~		

6) Draw chart of Data Type Classification.

- The type, or data type, or a variable determines a set of values that a variable might take & a set of operations that can be applied to those values.
- Data types can be broadly classified as...

→ Data Type :

- Primitive / Basic Data Type
  - char
  - int
  - float
  - double

- Derived Data Type
  - array
  - function
  - pointer

- User-defined Data Type
  - Structure
  - union
  - enumeration

- Valueless
  - void

7) What do you mean by Size & Sign Qualifiers? Explain with example.

→ Size specifier - short & long
Sign specifier - signed & unsigned
Type qualifier - constant volatile & restrict
The minimum size of a short int is 2 byte
The size of an int must be greater than or equal to that of an int.
The size of a long int must be greater than or equal to that of an int.
The minimum size of a long int is 4 bytes

- The most of the DOS based compilers that work on 16-bit computers, the size of a short int & an int is the same, which is two bytes.

- In 32 bit machine compilers such as GND C an int & long int take 4 bytes while a short int occupies 4 bytes

	16-bit	32-bit	64-bit
short int	2	2	2
int	2	4	4
long int	4	4	8

C99 provides two additional integer types long longint & unsigned long log int
For long long, the C99 standard specified at least 64 bit to support.

8) List operators available in C language.

- An operator is a symbol that specifies the mathemetical, logical or relational operation to be performed.

Operators

- Arithmetical (Unary / Binary / Ternary)
- Equality
- Relational
- Logical
- Bitwise
- Assignment (Simple / Compound / Expression)

9) What is lvalue & rvalue ?

- An lvalue is an expression to which a value can be assigned. An rvalue can be defined as an expression that can be assigned to an lvalue.

- The lvalue expression is located on the left side of an assignment statement whereas an rvalues is located on the right side of an assignment statement.

- The address associated with a program variable in C is called lvalues. The content of that location are its rvalues the quantity that is supposed to be the value of the value of the variable.

- The value of variable may change as program execution proceeds but never its lvalue.

- The distinction betn lvalues & rvalues becomes sharper if one considers the assignment operation with variable a & b, a=b.

- b, on the right hand side of the assignment operator, is the quantity to be found at the address associated with b i.e on rvalue a is assigned the value stored in the address at which the content are altered as a result of the assignment.

- a is an lvalue the assignment operation stores b's rvalue at a's lvalue.