Introduction to C Programming

C programming is a language designed for interacting with computers. C is a structural language. Structured language facilitates the development of a variety of programs in small modules. Lengthy programs are divided into short programs. Structured languages are more accessible than non-structured languages like B and COBOL. The C programming language is also called a system-programming language because it is beneficial for writing operating systems, compilers, interpreters, and network drivers.

Programming is the core of the software that runs all over the devices. The applications we use on smartphones, email applications, and even Google search engine functions are all written in codes. C programming is the most basic level of the program that is elemental in creating interactive and easy programs. Going by the programming in C book, you must know the different elements of the C program and the type of data structures used in it. Programming in C GATE CS questions comes in many exams, including CAT, GATE CS, PDU CS, SDE interviews, and more. Hence, it is advisable to know the programming in C questions from the specially designed GATE notes so that it is helpful for you in every job aspect. Programming in C online tests is also part of various state and national-level exams.

Many programming languages exist besides C, but C programming is the oldest and most widely used language today. Some of the features of C programming are as follows:

- The C programming language contains modules called functions
- The C functions are considered the basic building blocks.
- In C, one can develop and execute a program fast.
- It is a structural language like PASCAL and FORTRAN.
- It is suitable for both application and system software.

What is a Low-Level Language?

A low-level language is one in which instructions are given in machine codes, and a machine language is one that only understands 0 (zero) and 1 (one). The most famous low-level programming languages are C and C++. Low-level languages are further divided into two:

1. Machine Language
2. Assembly Language

Basics of C Programming

Every programming language has its set of valid symbols that are termed differently. A program is a set of instructions for performing a particular task. These instructions are just like English commands. A written program is called a source program. The source program is to be converted into the machine language. In C programming the source code is translated using a compiler.

Before jumping onto the structure of the C language, we need to understand the prerequisites. To start with the structure of the C program, let's first discuss tokens, Identifiers, keywords, data types, and functions. These are considered to be the basic building elements of any C program.

Token in C Programming

A token is defined as the smallest individual unit in C Programming. In C language, we have five different types of tokens, namely: keywords, identifiers, operators, special symbols, and constants.

- **Keyword in C Programming:** A keyword in C is defined as a reserved word that cannot be used as an identifier. C language has 32 keywords. The meaning of a keyword is predefined. Example: brake, char, int, continue, default, do, etc.
- **An Identifier in C Programming:** An identifier in C programming is a name provided to a variable in C, function, etc., other than a keyword. An identifier can be any valid string from the English language.
- **Constants:** Fixed values that do not change during the execution of a C program. Example: 100 is an integer constant, 'a' is a character constant, etc.
- **Operators**: The operator is a symbol that indicates and instructs the computer to perform mathematical or logical calculations. Examples: Arithmetic operators, Conditional operators, Bitwise operators, Logical operators, etc.
- **Delimiters / Separators:** These are used to separate variables, statements, and constants. Examples: commas, semicolons, double quotes, apostrophes, blank spaces, etc.
- **Strings**: String constants are specified in double quotes. "Exampgate exameexam" is a string constant as it is inside double quotes.

Define Data Types in C Programming

The data type is one of the most important attributes of an identifier. It determines the possible values that an identifier can take. In C language, data types are broadly classified into the following types:

- Basic data types
- Derived data types
- User-defined data types

It is important for candidates to know the difference between fundamental and derived data types in C for the GATE exam.

Function in the C Language: A function in C programming is defined as a module that can be re-utilized by the user or programmer once created. Every function in C has its definition, declaration, and body. In C, the critical function considered the heart of the program is the main().

Different Types of Modifiers with their Range

Types of Modifier	Size (In Bytes)	Range of Values
Int	2	-32768 to +32767
Signed Int	2	-32768 to +32767
Unsigned Int	2	0 to 65535
Short Int	2	-32768 to +32767
Long Int	4	-2147483648 to +2147483647
Float	4	-(3.4E + 38) to +(3.4E + 38)
Double	8	-(1.7E + 308) to +(1.7E + 308)
Char	1	-128 to +127
Unsigned Char	1	0 to 255

Structure of a C Program

Knowing the basics, we can now discuss the structure of a C program. In general, a C program is composed of three sections, which are as follows:

1. Pre-processor directives
2. Global declarations
3. Functions

The basic and simple C program structure includes all the sections mentioned above, data types, variables, functions, etc. The C program structure is as follows:

```
#include<stdio.h>

main()

{

int height= 10;

printf("%d”, height);

}
```

In the above example pre-processor directive is “ #include<stdio.h>”, the program starts with function “main()”, another function used is printf(), height is the variable of type integer.

Types of Operators in C Program

Operators	Symbol
Arithmetic operators	+, -, *, /, %, ++, --
Assignment operator	=, +=, -=, *=, etc
Relational operators	<, <=, >, >=, !=, ==
Logical operators	&&, ||, !
Bitwise operators	&, |, ~, ^, <<, >>
Special operators	sizeof(), comma, →
Pointer operators	* - Value at address (indirection), & - Address Operator

Type Conversions

Implicit Type Conversion: There are certain cases in which data will get automatically converted from one type to another:

  - When data is being stored in a variable, the data being stored does not match the type of the variable.
  - The data is stored will be converted to match the type of the storage variable.
- When an operation is performed on data of two different types, the "smaller" data type will be converted to match the "larger" type.
- The following example converts the value of x to a double-precision value before performing the division. Note that if the 3.0 were changed to a simple 3, integer division would be performed, losing any fractional values in the result.
  - average = x / 3.0;
  - When data is passed to or returned from functions.

Explicit Type Conversion: Data may be expressly converted using the typecast operator.

- The following example converts the value of x to a double-precision value before performing the division. (y will then be implicitly promoted, following the guidelines listed above.)
  - average = (double) x / y;
  - Note that: x itself is unaffected by this conversion.

Control Statements in C Program

A control statement in C is one of the instructions, statements, or groups of statements in a programming language that determines the execution of other statements or instructions. C provides two types of flow controls.

1. Looping (decides how many times to perform a certain action)
2. Branching (deciding what action to take)

If Statement

It takes an expression inside the parenthesis and a statement in it. Expressions are assumed to be true if evaluated values are non-zero. Syntax of if Statement:

(i)

```
if (condition) statement
```

(ii)

```
if (condition)
{
statement1
statement2
:
}
```

If Else Statement

```
if (condition)

{ statements }

else { statements }
```

The switch Statement

The switch statement tests the given variable's value (or expression) against a list of a few case values. If a case is matched, the block of statements associated with that case is executed:

```
switch (control variable)

{

case constant-1: statement(s);

break;

case constant-2: statement(s);

break;

:

case constant-n: statement(s);

break;
default: statement(s);

}
```

The Conditional Operators (? :):

The ?: operators are like an if-else statement except that we can use it within expressions because it is an operators. ?: are ternary operators in that it takes three values. They are the only ternary operator in the C language.

```
flag = (x < 0) ? 0 : 1;
```

This conditional statement can be evaluated as following with an equivalent if else statement.

```
if (x < 0) flag = 0;
 else flag = 1;
```

Loop Control Structure

Loops are the way to repeat controls and commands. This involves repeating a portion of the program several times until a particular condition is satisfied.

- **while Loop:**

```
initialize loop counter;
while (test loop counter using a condition/expression)

{

<Statement1>

<Statement2>

...

< decrement/increment loop counter>

}
```

- **for Loop:**

```
for (initialize counter; test counter; increment/decrement counter)
{

<Statement1>

<Statement2>

...

}
```

- **do while Loop:**

```
initialize loop counter;
do
```

```
{
<Statement1>
<Statement2>
...
}
while (this condition is true);
```

Important Programming in C Topics for GATE CS

Programming is a broad topic and has vast syllabi. The major topics for programming in C study material for GATE CS are given below:

Topic	Explanation
Variables and Declarable	Variables are anything whose value is not fixed. We assign some values to it using different kinds of data structures. Declarations are the initial step of defining any variable or function.
Functions	Functions are the real operations that we do in programming. We can perform basic functions such as addition and subtraction to complex functions like segregating any data.
Arrays and Strings	There is a data structure that stores values in different formats.

Tips to Solve Programming in C Questions in GATE CS Exam

You can follow the following tips while solving programming in C questions. Prepare for the topic by solving last year's question papers well.

- Programming in C GATE questions and answers PDF are based on the programming in C and their uses in different scenarios. Hence you should focus on the programming section for programming topics.
- Prepare excellent programming in C notes for the GATE that contains all the details of the syntactic analysis. Those programming in C GATE exam notes will help you in revisions.
- Practice programming in C online quizzes from online sites. These online tests help you in preparation for the real exam.
- Be clear with the programming in C topics for the GATE CSE exam as many repetitive topics, such as basic C programs using recursions, data structure, and different imported elements.

Importance of Programming in C in GATE

Here's why the tree is an integral part of the GATE and other competitive exams:

- It checks whether you can frame a workable program in the C language. It also checks your grasp of the programming in the C GATE CSE syllabus.
- The weightage of programming in the C GATE topic is 3% to 10% in the GATE Exam.
- GATE requires you to acquire basic knowledge of programming and functions. It helps to maintain the accuracy of any main program.
- The concept of programming in C GATE is used in real-life situations as a software engineer solving various programming problems.

Most Recommended Books for Programming in C for GATE

The following are some excellent books on programming in C for computer science that can help you prepare for GATE's Programming in C syllabus.

Book Title	Author
The C Programming Language	Brian K.
C Programming	Kim N. King
Expert C Programming	Peter Van Der Linden

Why You Should Prepare Programming in C from BYJU'S Exam Prep?

BYJU'S Exam Prep is an online source of knowledge that covers excellent quality tutorials and questions regarding the topic. You can get all the preparatory materials like programming in C quiz, programming in C MCQ PDF, programming in C MCQ questions, and more from BYJU'S Exam Prep website. These preparatory materials are a valuable source to test your exam preparation levels and boost your main exam performance. To prepare well for the exams, our experienced educators have designed programming in C notes for CS and programming in C notes for GATE PDF students who help you revise.