Que. What is the main goal of coadding?

Naming convention in c programming:

# Naming Convention in C Programming

###### **C Basic Concepts**

In C programming, naming conventions are set of rules for choosing the valid name to be used for variables and functions in a C program.

### Naming Conventions rules for Variables and Methods (Functions) are:

* It should begin with an alphabet.
* There may be more than one alphabet, but without any spaces between them.
* Digits may be used but only after alphabet.
* No special symbol can be used except the underscore (\_) symbol. When multiple words are needed, an underscore should separate them.
* No keywords or command can be used as a variable name.
* All statements in C language are case sensitive. Thus a variable A (in uppercase) is considered different from a variable declared a (in lowercase).

Now let's see some examples for more understanding.

#### Example 1

It should begin with an alphabet.

x // x is a valid variable name because it starts with an alphabet x

#### Example 2

There may be more than one alphabet, but without any spaces between them.

total // total is a valid variable name as there is no space between alphabets

#### Example 3

Digits may be used but only after alphabet.

ar15 // ar15 is a valid variable name as digits have been used after the alphabet

a6b2 // a6b2 is a valid variable name as digits have been used after the alphabet

#### Example 4

No special symbol should be present within the variable name except underscore \_.

total\_cost // total\_cost is a valid variable name as there is an underscore

total cost // total cost is an invalid variable name as there is a space

total-cost // total-cost is an invalid variable name as there is a hyphen

total$ // total$ is an invalid variable name as there is a dollar symbol

#### Example 5

No keywords or command can be used as a variable name.

for // here for is an invalid variable name because it is a keyword in C

if // if is an invalid variable name because it is a keyword in C

case // case is an invalid variable name because it is a keyword in C

const // const is an invalid variable name because it is a keyword in C

#### Example 6

All statements in C language are case sensitive. Thus a variable A (in uppercase) is considered different from a variable declared a (in lowercase).

a // a is a valid variable written in lowercase

A // A is a valid variable written in uppercase so both are different variables

Ans. The aim of the coding is **to produce high quality system which can be performed in any**

**situation**. The programmer removes all the errors related to syntax and format and all the logical errors which find in the programmer during the coding phase. Some of the objectives are defined below in concerned to coding.

Unit 2:

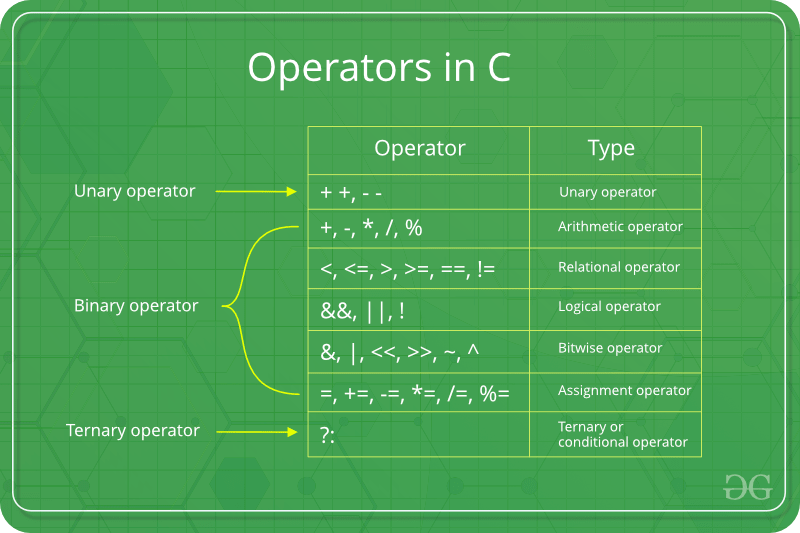
Que. What is the meaning of type casting and type conversion in C programming?

Ans.

**Difference Between Type Casting and Type Conversion**

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Type Casting** | **Type Conversion** |
| 1 | Type casting is a mechanism in which one data type is converted to another data type using a casting () operator by a programmer. | Type conversion allows a compiler to convert one data type to another data type at the compile time of a program or code. |

1. operator:

Ans: 

Que. Can we define a function in header file:

Ans. Yes....

A header file is a file with extension **.h** which contains C function declarations and macro definitions to be shared between several source files. There are two types of header files: the files that the programmer writes and the files that comes with your compiler.

You request to use a header file in your program by including it with the C preprocessing directive **#include**, like you have seen inclusion of **stdio.h** header file, which comes along with your compiler.

Including a header file is equal to copying the content of the header file but we do not do it because it will be error-prone and it is not a good idea to copy the content of a header file in the source files, especially if we have multiple source files in a program.

A simple practice in C or C++ programs is that we keep all the constants, macros, system wide global variables, and function prototypes in the header files and include that header file wherever it is required.

Include Syntax

Both the user and the system header files are included using the preprocessing directive **#include**. It has the following two forms −

#include <file>

This form is used for system header files. It searches for a file named 'file' in a standard list of system directories. You can prepend directories to this list with the -I option while compiling your source code.

#include "file"

This form is used for header files of your own program. It searches for a file named 'file' in the directory containing the current file. You can prepend directories to this list with the -I option while compiling your source code.

Include Operation

The **#include** directive works by directing the C preprocessor to scan the specified file as input before continuing with the rest of the current source file. The output from the preprocessor contains the output already generated, followed by the output resulting from the included file, followed by the output that comes from the text after the **#include** directive. For example, if you have a header file header.h as follows −

char \*test (void);

and a main program called *program.c* that uses the header file, like this −

int x;

#include "header.h"

int main (void) {

puts (test ());

}

the compiler will see the same token stream as it would if program.c read.

int x;

char \*test (void);

int main (void) {

puts (test ());

}

Once-Only Headers

If a header file happens to be included twice, the compiler will process its contents twice and it will result in an error. The standard way to prevent this is to enclose the entire real contents of the file in a conditional, like this −

#ifndef HEADER\_FILE

#define HEADER\_FILE

the entire header file file

#endif

This construct is commonly known as a wrapper **#ifndef**. When the header is included again, the conditional will be false, because HEADER\_FILE is defined. The preprocessor will skip over the entire contents of the file, and the compiler will not see it twice.

Computed Includes

Sometimes it is necessary to select one of the several different header files to be included into your program. For instance, they might specify configuration parameters to be used on different sorts of operating systems. You could do this with a series of conditionals as follows −

#if SYSTEM\_1

# include "system\_1.h"

#elif SYSTEM\_2

# include "system\_2.h"

#elif SYSTEM\_3

...

#endif

But as it grows, it becomes tedious, instead the preprocessor offers the ability to use a macro for the header name. This is called a **computed include**. Instead of writing a header name as the direct argument of **#include**, you simply put a macro name there −

Que. Declaration of function in header file vs normal declaration of function in c programming?

Ans.

Que. What is the meaning of .h in c programming?

Ans. Advertisements. **A header file** is a file with extension . h which contains C function declarations and macro definitions to be shared between several source files. There are two types of header files: the files that the programmer writes and the files that comes with your compiler.

Que. What is meaning of ternary operator in c programming?

## Ans. Syntax of Ternary Operator

The syntax of ternary operator is :

testCondition ? expression1 : expression 2;

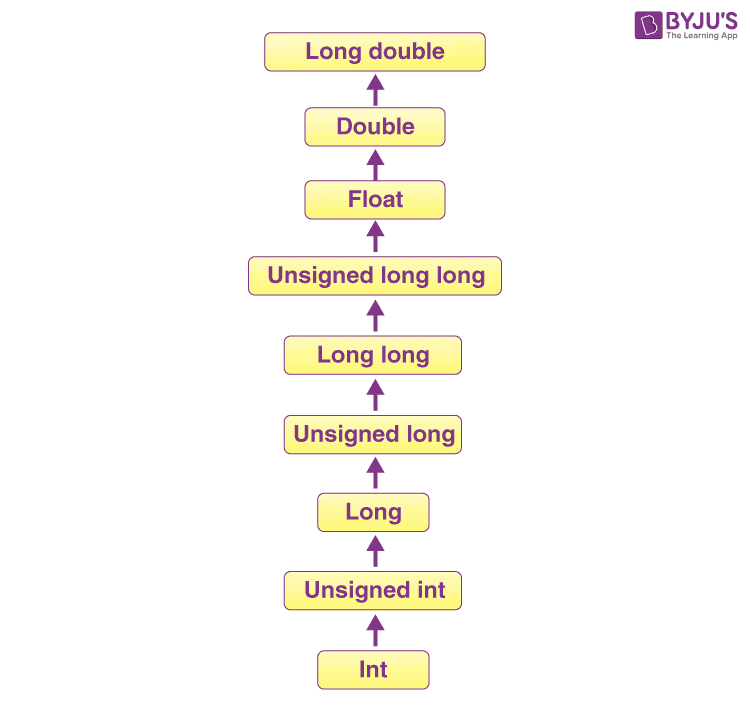
The testCondition is a boolean expression that results in either **true** or **false**. If the condition is

* true - **expression1** (before the colon) is executed
* false - **expression2** (after the colon) is executed

The ternary operator takes 3 operands (condition, expression1 and expression2). Hence, the name **ternary operator**.

Que. Conversion and type casting of C program?

Ans.



Que. What is the size of int, char, float data type?

Ans. C/C++ program to find the size of int, float, double and char

|  |  |  |
| --- | --- | --- |
| **Data Type** | **Memory (bytes)** | **Range** |
| **signed char** | **1** | **-128 to 127** |
| unsigned char | 1 | 0 to 255 |
| float | 4 |  |
| double | 8 |  |

Que. What is the meaning of binary operator in c programming?

Ans. A binary operator is **an operator that operates on two operands and manipulates them to return a result**. Operators are represented by special characters or by keywords and provide an easy way to compare numerical values or character strings.