C- TypeCasting

Typecasting in C is the process of converting one data type to another data type by the programmer using the casting operator during program design.

In typecasting, the destination data type may be smaller than the source data type when converting the data type to another data type, that's why it is also called narrowing conversion.

Syntax:

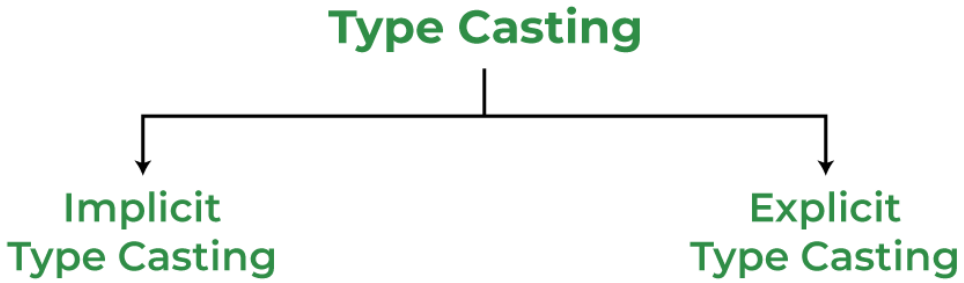
```
int x;

float y;

y = (float) x;
```

Types of Type Casting in C

In C there are two major types to perform type casting.

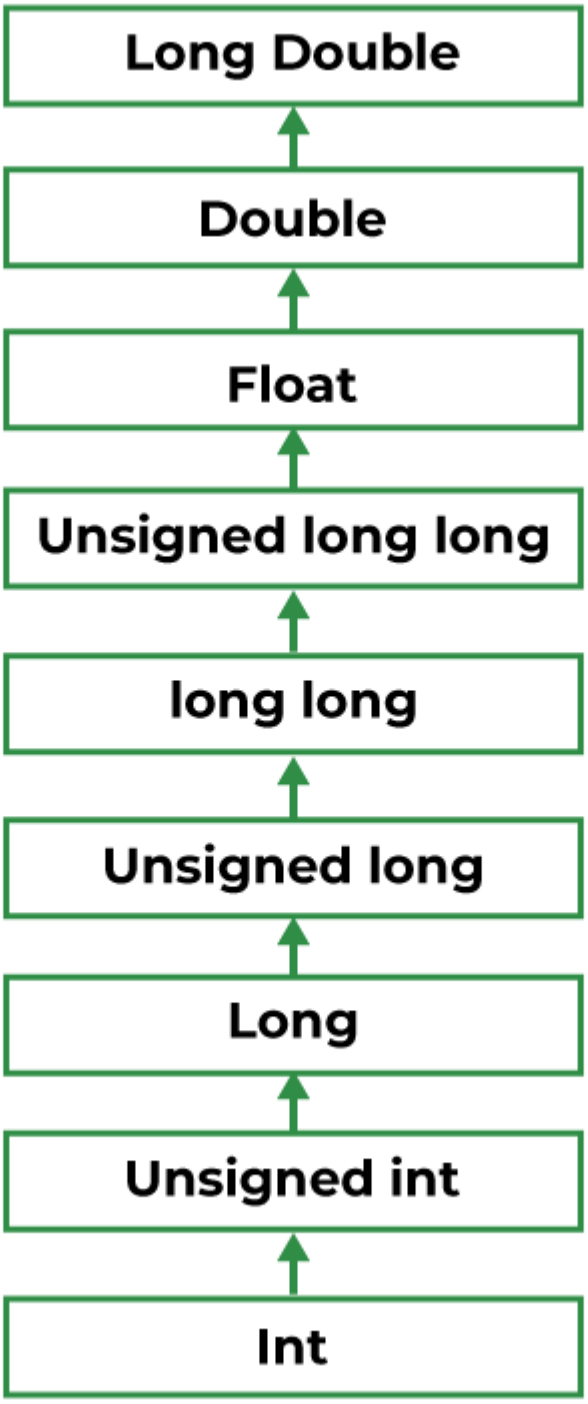


- Implicit type casting
- Explicit type casting

1. Implicit Type Casting

Implicit type casting in C is used to convert the data type of any variable without using the actual value that the variable holds. It performs the conversions without altering any of the values which are stored in the data variable. Conversion of lower data type to higher data type will occur automatically.

Integer promotion will be performed first by the compiler. After that, it will determine whether two of the operands have different data types. Using the hierarchy below, the conversion would appear as follows if they both have varied data types:



2. Explicit Type Casting

There are some cases where if the datatype remains unchanged, it can give incorrect output. In such cases, typecasting can help to get the correct output and reduce the time of compilation. In explicit type casting, we have to force the conversion between data types. This type of casting is explicitly defined within the program.

Program 1:

- C

```c
// C program to illustrate the use of
// typecasting
#include <stdio.h>

// Driver Code
int main()
{
    // Given a & b
    int a = 15, b = 2;
    float div;

    // Division of a and b
    div = a / b;

    printf("The result is %f\n", div);

    return 0;
}
```

Output:

```
The result is 7.000000
```

Explanation: Here, the actual output needed is **7.500000**, but the result is **7.000000**. So to get the correct output one way is to change the data type of a given variable. But

correct output can also be done by **typecasting**. This consists of putting a pair of parentheses around the name of the data type like **division = (float) a/b**.

Program 2:

- C

```
// C program to showcase the use of

// typecasting

#include <stdio.h>



// Driver Code

int main()

{

    // Given a & b

    int a = 15, b = 2;

    char x = 'a';



    double div;



    // Explicit Typecasting in double

    div = (double)a / b;



    // converting x implicitly to a+3 i.e, a+3 = d

    x = x + 3;
```

```
    printf("The result of Implicit typecasting is %c\n", x);


    printf("The result of Explicit typecasting is %f", div);


    return 0;
}
```

Output

```
The result of Implicit typecasting is d

The result of Explicit typecasting is 7.500000
```

Explanation: In the above C program, the expression **(double)** converts variable a from type **int** to type **double** before the operation.
In C programming, there are 5 built-in type casting functions.

- **atof():** This function is used for converting the string data type into a float data type.
- **atbol():** This function is used for converting the string data type into a long data type.
- **Itoa():** This function is used to convert the long data type into the string data type.
- **itoba()**: This function is used to convert an int data type into a string data type.
- **atoi()**: This data type is used to convert the string data type into an int data type.

Advantages of Type Casting

- Type casting in C programming makes the program very lightweight.
- Type representation and hierarchies are some features we can take advantage of with the help of typecasting.
- Type casting helps programmers to convert one data type to another data type.