Type casting is a way to convert a variable from one data type to another data type. For example, if you want to store a long value into a simple integer then you can type cast long to int. You can convert values from one type to another explicitly using the cast operator.

New data type should be mentioned before the variable name or value in brackets which to be typecast.

**C type casting example program:**

In the below C program, 7/5 alone will produce integer value as 1. So, type cast is done before division to retain float value (1.4).

```c
#include <stdio.h>

int main ()
{
    float x;
    x = (float) 7/5;
    printf("%f",x);
}
```

**Output:**

1.400000

What is type casting in C Language?

Converting an expression of a given type into another type is known as type-casting. Typecasting is more use in C language programming.
Here, it is best practice to convert lower data type to higher data type to avoid data loss.

Data will be truncated when higher data type is converted to lower. For example, if float is converted to int, data which is present after decimal point will be lost.

**There are two types of type casting in c language.**

**Types of typecasting in C**

<table>
<thead>
<tr>
<th>S.No</th>
<th>Types of typecasting in C Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Implicit Conversion</td>
</tr>
<tr>
<td>2</td>
<td>Explicit Conversion</td>
</tr>
</tbody>
</table>

**1. Implicit conversion**

Implicit conversions do not required any operator for converted. They are automatically performed when a value is copied to a compatible type in program.

Here, the value of a has been promoted from int to double and we have not had to specify any type-casting operator. This is known as a standard conversion.

**Example :-**

```c
#include<stdio.h>

#include<conio.h>

void main()
{
    int i=20;
}
```
2. Explicit conversion

In c language, Many conversions, specially those that imply a different interpretation of the value, require an explicit conversion. We have already seen two notations for explicit type conversion.

They are not automatically performed when a value is copied to a compatible type in program.

Example :-

```c
#include<stdio.h>
#include<conio.h>
void main()
{
    int i=20;
    short p;
    clrscr();
}
```
p = (short) i; // Explicit conversion

printf("Explicit value is %d",p);

getch();

Output :-
Explicit value is 20.

**Usual Arithmetic Conversion**

The usual arithmetic conversions are implicitly performed to cast their values in a common type, C uses the rule that, in all expressions except assignments, any implicit type conversions made from a lower size type to a higher size type as shown below:
Inbuilt Typecast Functions In C:

There are many inbuilt typecasting functions available in C language which performs data type conversion from one type to another.
We are recently taking a survey from different programmers who are available in google plus social media. We did a survey for type casting in c language.

<table>
<thead>
<tr>
<th>S.No</th>
<th>TypeCast Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>atof()</td>
<td>Convert string to Float</td>
</tr>
<tr>
<td>2</td>
<td>atoi()</td>
<td>Convert string to int</td>
</tr>
<tr>
<td>3</td>
<td>atol()</td>
<td>Convert string to long</td>
</tr>
<tr>
<td>4</td>
<td>itoa()</td>
<td>Convert int to string</td>
</tr>
<tr>
<td>5</td>
<td>ltoa()</td>
<td>Convert long to string</td>
</tr>
</tbody>
</table>