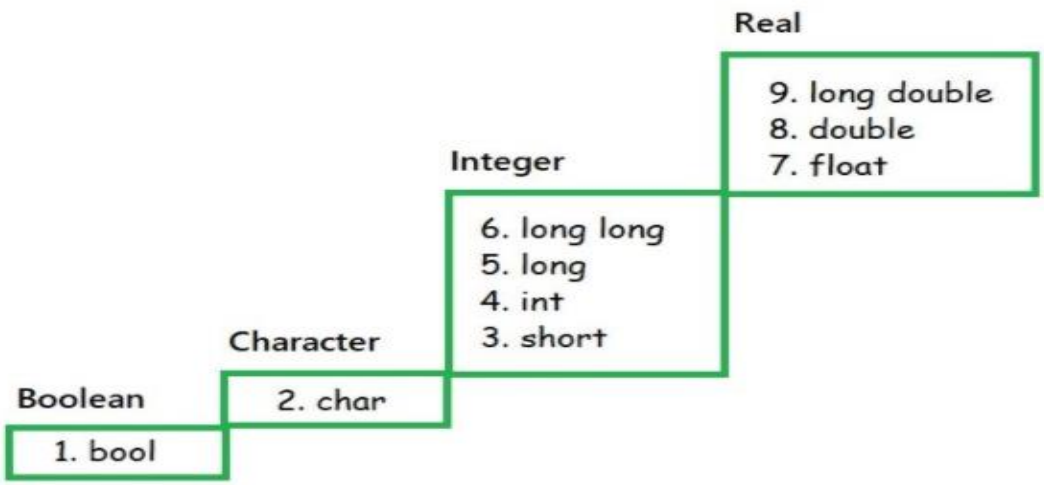
Suppose, you have a variable whose value is 23.2332 but at some line of your code, you want to use its integer value (floor value) only i.e., 23. The simplest solution is **type casting.**

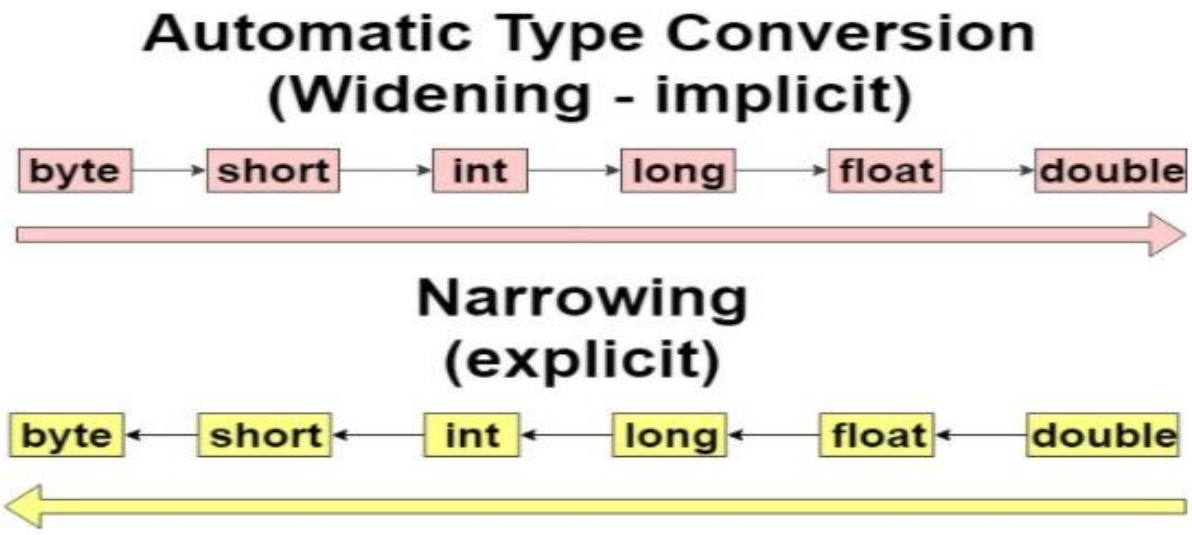
Converting one datatype into another is known as type casting or, type-conversion.

For example, if you want to store a 'long' value into a simple integer then you can type cast 'long' to 'int'.

There are two types of type conversion:

1. Implicit Type Conversion
2. Explicit Type Conversion





Implicit Type Conversion

Implicit type casting means conversion of data types without losing its original meaning. Implicit type conversion happens automatically when a value is copied to its compatible data type. During conversion, strict rules for type conversion are applied. If the operands are of two different data types, then an operand having lower data type is automatically converted into a higher data type.

Also known as 'automatic type conversion'.

1. Done by the compiler on its own, without any external trigger from the user.
2. Generally, takes place when in an expression more than one data type is present. In such condition type conversion (type promotion) takes place to avoid lose of data.
3. All the data types of the variables are upgraded to the data type of the variable with largest data type.

   bool -> char -> short int -> int ->
   unsigned int -> long -> unsigned ->
   long long -> float -> double -> long double
4. It is possible for implicit conversions to lose information, signs can be lost (when signed is implicitly converted to unsigned), and overflow can occur (when long long is implicitly converted to float).

main.c

```c
/*******************************************
Statement - implicit conversion
Written For - PRE-CAT COURSE by CCATPREPARATION.COM
*******************************************/

#include<stdio.h>
int main()
{
    int x = 10; // integer x
    char y = 'a'; // character c

    // y implicitly converted to int. ASCII
    // value of 'a' is 97
    x = x + y;

    // x is implicitly converted to float
    float z = x + 1.0;

    printf("x = %d, z = %f", x, z);
    return 0;
}
```

input

```
x = 107, z = 108.000000
```

Converting Character to Int

Consider the example of adding a character decoded in ASCII with an integer:

an integer promotion by converting the value of 'k' to ASCII before performing the actual addition operation.

main.c

```c
/*****************************************
Statement - implicit conversion
Written For - PRE-CAT COURSE by CCATPREPARATION.COM
*****************************************/

#include <stdio.h>

main() {
   int  number = 1;
   char character = 'k'; /*ASCII value is 107 */
   int sum;
   sum = number + character;
   printf("Value of sum : %d\n", sum );
}
```

input

```
main.c:8:1: warning: return type defaults to 'int' [-Wimplicit-int]
Value of sum : 108
```

Explicit Type Conversion

In implicit type conversion, the data type is converted automatically. There are some scenarios in which we may have to force type conversion. Suppose we have a variable div that stores the division of two operands which are declared as an int data type.

```
int result, var1=10, var2=3;
result=var1/var2;
```

In this case, after the division performed on variables var1 and var2 the result stored in the variable "result" will be in an integer format. Whenever this happens, the value stored in the variable "result" loses its meaning because it does not consider the fraction part which is normally obtained in the division of two numbers.

To force the type conversion in such situations, we use explicit type casting.

It requires a type casting operator. The general syntax for type casting operations is as follows: `(type-name) expression`

main.c

```c
/*****************************************
Statement - Explicit conversion
Written For - PRE-CAT COURSE by CCATPREPARATION.COM
*****************************************/

#include<stdio.h>
int main()
{

    double x = 1.2;

    // Explicit conversion from double to int
    int sum = (int)x + 1;

    printf("sum = %d", sum);

    return 0;


    return 0;
}
```

Summary

- Typecasting is also called as type conversion
- It means converting one data type into another.
- Converting smaller data type into a larger one is also called as type promotion.
- 'C' provides an implicit and explicit way of type conversion.
- Implicit type conversion operates automatically when the compatible data type is found.

- Explicit type conversion requires a type casting operator.

Keep in mind the following rules for programming practice when dealing with different data type to prevent from data loss:

- Integers types should be converted to float.
- Float types should be converted to double.
- Character types should be converted to integer.