

Type Casting

By: Annamalai A

Type casting simply means to convert the data types of variables. It may sound really cool, and it surely is and is also really useful, but it must be used with caution because there are chances for data loss. Let's dive deep into how change types of variables and what programs to internally.

There are two ways that variables change their types:

- **Implicit Type Casting:** The compiler changes the data type of variables due to certain operations.
- **Explicit Type Casting:** The programmer changes the data type of variables manually.

Implicit Type Casting

Also called type promotion, the data types of variables are changed by the compiler due to certain operations. The order usually goes like this

char → int → float → double

The order is like this because each of these types store more data from left to right. The number of bytes they store increase from left to right. However, the reverse order may not occur due to *data loss*.

Data loss is basically when some information stored by variable gets lost due to unexpected events (type casting in this case). A `double` stores 8 bytes of information whereas `float` stores only 4 bytes. When `double` is converted to `float`, the number gets rounded off and 4 bytes of information goes missing.

```
#include <stdio.h>

int main() {
    int a = 5;
    float b = 2.5;
    float result = a + b; // int 'a' is promoted to float
```

```
    printf("Result = %f\n", result);
    return 0;
}
```

Explicit Type Casting

This is done explicitly .i.e. Manually by the user. The syntax to do this is

(*<new_type>*) *<variable>* .

```
#include <stdio.h>

int main() {
    int a = 10, b = 3;
    float div = (float)a / b; // explicitly cast 'a' to float
    printf("Division = %.2f\n", div);
    return 0;
}
```