

# type conversion in C++

---

## ◆ Type Conversion in C++

### ✓ Definition:

**Type conversion** (or type casting) is the process of converting a value from one data type to another.

There are two main types:

1. **Implicit Conversion (Type Promotion / Coercion)**
  2. **Explicit Conversion (Type Casting)**
- 

## ◆ 1. Implicit Type Conversion (Automatic)

The compiler automatically converts one data type to another when necessary, usually **from lower to higher data type** to prevent data loss.

### 📌 Examples of Implicit Conversion:

- `char` to `int`
- `int` to `float`
- `float` to `double`

### ✓ Example:

```
#include <iostream>
using namespace std;
```

```
int main() {
    int a = 10;
    float b = 5.5;
```

```
    float result = a + b; // 'a' (int) is implicitly converted to float
    cout << "Result: " << result << endl; // Output: 15.5
```

```
    return 0;
}
```

In the above, `a` is implicitly converted to `float` to match the type of `b`.

---

## ◆ 2. Explicit Type Conversion (Type Casting)

The programmer **forcibly** converts one data type to another using **casting operators**.

### ✓ Syntax:

(type) variable

### 📌 Common Types of Explicit Casting:

- `int(x)` or `(int)x`
  - `float(x)`
  - `double(x)`
  - `static_cast<T>(x)` – recommended in C++
- 

### ✓ Example: Basic C-style Casting

```
#include <iostream>
using namespace std;

int main() {
    double price = 99.99;
    int roundedPrice = (int)price; // Explicitly cast to int

    cout << "Rounded Price: " << roundedPrice << endl; // Output: 99

    return 0;
}
```

---

### ✓ Example: Using `static_cast<T>()` (Modern C++ Style)

```
#include <iostream>
using namespace std;

int main() {
    int a = 5, b = 2;
    double result = static_cast<double>(a) / b;

    cout << "Result: " << result << endl; // Output: 2.5

    return 0;
}
```

Without `static_cast`, result would be integer division (2), not 2.5.

---

## ◆ Summary Table

Type	Triggered by	Example	Risk of Data Loss
Implicit	Compiler	<code>int x = 5; float y = x + 1.5;</code>	No (usually safe)
Explicit	Programmer	<code>int x = (int)3.14;</code> or <code>static_cast&lt;int&gt;(3.14)</code>	Yes (programmer takes responsibility)

---

## ◆ Bonus: User-Defined Type Conversion

You can overload type conversion in your own classes.

### ✅ Example:

```
#include <iostream>
using namespace std;

class Distance {
    int meters;
public:
    Distance(int m) : meters(m) {}

    // User-defined conversion to int
    operator int() {
        return meters;
    }
}
```

```
};
```

```
int main() {  
    Distance d(100);  
    int m = d; // Calls user-defined conversion operator  
    cout << "Meters: " << m << endl;  
  
    return 0;  
}
```

---