Type-Casting

In this lesson, you will learn about converting one data type to another.

We'll cover the following

- Introduction
- Types of casting
  - Implicit casting
  - Example program
  - Explicit casting
  - Example program

Introduction

Suppose you have initialized a variable with a `char` data type. At some point in a program, you need its integer value. For such situations, type-casting comes in.

Type-casting is a way to convert the value of one data type to another data type.

Types of casting

Type-casting has two types:

- Implicit casting
- Explicit casting

Implicit casting

In implicit casting, the compiler automatically converts one data type to another.

For example, if you store a floating-point value into a variable of integer type, the compiler will convert the `float` value into `int` without any user intervention.

Example program

Run the code below and see the output!

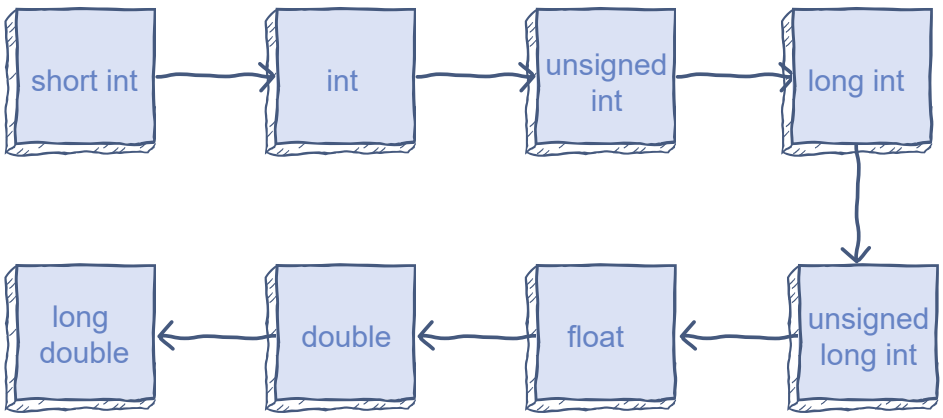
```
#include <iostream>

using namespace std;

int main() {
  // Stores floating point value in variable of type int
  int int_value = 13.9;
  // Displays the value of variable
  cout << int_value;

}
```

Implicit casting

In the code above, we have stored a floating-point value in the variable of type `int`. The compiler automatically knows that it should truncate the value after the decimal point. Therefore, it stores `13` in the variable `int_value`.

We should always do the type-casting from smaller to the larger data types. Otherwise, you can lose your data. For example, in the above program, we are losing the information after the decimal point.

The arrows in the figure given below show the order in which we can do the conversion without any data loss. For example, we can convert `short int` into `int` without any loss of data or precision.



Explicit casting

In explicit casting, the user manually converts one data type to another. The basic syntax for explicit type casting in C++ is:

```
( data type ) variable_name ;
```

Example program

Suppose you want to know the ASCII value of a character stored in a variable. Let's write a program!

Press the **RUN** button and see the output!

```
#include <iostream>

using namespace std;

int main() {
  // Initializes a variable of char data type
  char character = 'A';
  // Declares a variable of int type
  int ASCII;
  // Converts char data type into int explicitly
  ASCII = (int) character;
  // Prints value of variable
  cout << "ASCII value = " << ASCII;

}
```

Explicit type casting

In the code above, **Line No. 10** takes a `char` data type on its right-hand side, converts into an `int`, and then stores the converted value into the variable on the left-hand side.

Quiz

What is the output of the following code?

```
int main() {
  int number = 90;
  char character = number;
  cout << "character = " << character;
}
```

See this link for the ASCII values

See this link for the ASCII values.

Retake Quiz

Let’s discuss strings and escape sequences in the upcoming lesson.