**Type Casting in Java**

In Java, **type casting** is a method or process that converts a data type into another data type in both ways manually and automatically. The automatic conversion is done by the compiler and manual conversion performed by the programmer. In this section, we will discuss **type casting** and **its types** with proper examples.



**Type casting**

Convert a value from one data type to another data type is known as **type casting**.

**Types of Type Casting**

There are two types of type casting:

* Widening Type Casting
* Narrowing Type Casting

**Widening Type Casting**

Converting a lower data type into a higher one is called **widening** type casting. It is also known as **implicit conversion** or **casting down**. It is done automatically. It is safe because there is no chance to lose data. It takes place when:

* Both data types must be compatible with each other.
* The target type must be larger than the source type.

1. **byte** -> **short** -> **char** -> **int** -> **long** -> **float** -> **double**

For example, the conversion between numeric data type to char or Boolean is not done automatically. Also, the char and Boolean data types are not compatible with each other. Let's see an example.

**WideningTypeCastingExample.java**

**public** **class** WideningTypeCasting;

{

**public** **static** **void** main(String[] args)

{

**int** x = 7;

//automatically converts the integer type into long type

**long** y = x;

//automatically converts the long type into float type

**float** z = y;

System.out.println("Before conversion, int value "+x);

System.out.println("After conversion, long value "+y);

System.out.println("After conversion, float value "+z);

}

}

**Output**

Before conversion, the value is: 7

After conversion, the long value is: 7

After conversion, the float value is: 7.0

In the above example, we have taken a variable x and converted it into a long type. After that, the long type is converted into the float type.

**Narrowing Type Casting**

Converting a higher data type into a lower one is called **narrowing** type casting. It is also known as **explicit conversion** or **casting up**. It is done manually by the programmer. If we do not perform casting then the compiler reports a compile-time error.

1. **double** -> **float** -> **long** -> **int** -> **char** -> **short** -> **byte**

Let's see an example of narrowing type casting.

In the following example, we have performed the narrowing type casting two times. First, we have converted the double type into long data type after that long data type is converted into int type.

**NarrowingTypeCastingExample.java**

**public** **class** NarrowingTypeCastingExample

{

**public** **static** **void** main(String args[])

{

**double** d = 166.66;

//converting double data type into long data type

**long** l = (**long**)d;

//converting long data type into int data type

**int** i = (**int**)l;

System.out.println("Before conversion: "+d);

//fractional part lost

System.out.println("After conversion into long type: "+l);

//fractional part lost

System.out.println("After conversion into int type: "+i);

}

}

**Output**

Before conversion: 166.66

After conversion into long type: 166

After conversion into int type: 166