TypeCasting in Java

TypeCasting in Java

TypeCasting in Java is the process of converting one primitive data type into another. TypeCasting can be done automatically and explicitly.
When we assign the value of one data type to another data type, then there is a chance that two data types might not be compatible with each other. The Java compiler will automatically perform the conversion if the data types are consistent. This type of conversion is known as Automatic Type Conversion. If the java compiler cannot perform the conversion automatically, they need to be cast explicitly.
There are two types of TypeCasting in Java.

- Widening or Automatic Type Conversion.
- Narrowing or Explicit Type Conversion.

1. Widening or Automatic Type Conversion: When we assign a value of a smaller data type to a large data type, this process is known as Widening Type Casting. It is also known as Automatic Type Conversion because the Java compiler will perform the conversion automatically. This can happen only when the two data types are compatible.

```
byte -> short -> int -> long -> float -> double ( Widening or Automatic Type Conversion
```

For example, In Java, int data types are compatible, but it isn't compatible with char and boolean data types. Also, char and boolean data types are not compatible with each other.

Example:

```
public class WideningConversation {
    public static void main(String args[]) {

        // Automatic Type Conversion.
        int i = 2147483647; // Int max value in java.
        long l = i; // Automatically converted to long, now we can extend l's value.
        l = l + 1;
        double d = l; // Automatically converted to double.
        System.out.println("Int value : " + i);
        System.out.println("Long value : " + l);
        System.out.println("Double value : " + d);
    }
}
```

Output

```
Int value : 2147483647
Long value : 2147483648
Double value : 2.147483648E9
```

2. Narrowing or Explicit Type Conversion: When we assign a value of a large data type to a small data type, the process is known as Narrowing Type Casting. This can't be done automatically. We need to convert the type explicitly. If we don't perform casting, the java compiler will give a compile-time error.

```
double -> float -> long -> int -> short -> byte  ( Narrowing or Explicit Type Conversio
```

Example:

```
public class ExplicitConversation {
        public static void main(String args[]) {

                // Explicit Type Conversion
                double d = 25.123;
                int i = (int) d;
                byte b = (byte) i;
                System.out.println("Double value : " + d);
                System.out.println("Int value : " + i);
                System.out.println("Byte value : " + b);
        }
}
```

Output

```
Double value : 25.123
Int value : 25
Byte value : 25
```