Type Casting in Java

Casting is a process of changing one type value to another type. In Java, we can cast one type of value to another type. It is known as type casting.

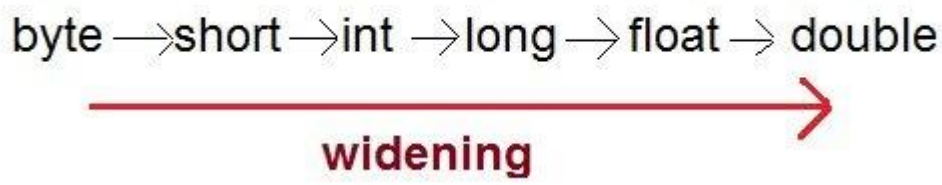
Example :

int x = 10;

byte y = (byte)x;

In Java, type casting is classified into two types,

Widening Casting(Implicit)



Narrowing Casting(Explicitly done)

Widening or Automatic type conversion

Automatic Type casting take place when, the two types are compatible the target type is larger than the source type.

Example:

```
class ImplicitTypeCasting
{
	public static void main(String[] args)
	{
		byte b = 100;
		short s = b;
		int i = b;
		long l = i;
		float f = l;
		double d = f;

		System.out.println("Byte value "+b);
		System.out.println("Short value "+s);
		System.out.println("Int value "+i);
		System.out.println("Long value "+l);
		System.out.println("Float value "+f);
		System.out.println("Double value "+d);
	}
}
```

Narrowing or Explicit type conversion

When you are assigning a larger type value to a variable of smaller type, then you need to perform explicit type casting. If we don't perform casting then compiler reports compile time error.

Example :

```
class ExplicitTypeCasting
{
	public static void main(String[] args)
	{
		double d = 100;
		float f = (float)d;
		long l = (long)f;
		int i = (int)l;
		short s = (short)i;
		byte b = (byte)s;

		System.out.println("Double value "+d);
		System.out.println("Float value "+f);
		System.out.println("Long value "+l);
		System.out.println("Int value "+i);
		System.out.println("Short value "+s);
		System.out.println("Byte value "+b);
	}
}
```