Day 08 – Type Conversion in Java

Type conversion means changing one data type into another. Java programs often need this because input, calculation, and output use different data types. Below is a simple explanation with examples.

1. Implicit Type Conversion (Widening)

In implicit conversion, Java automatically converts a smaller data type into a larger data type. There is no data loss.

Example:

```
int a = 10;
double b = a;
```

Here, int is converted to double automatically.

2. Explicit Type Conversion (Narrowing)

In explicit conversion, a larger data type is converted into a smaller data type. We must use type casting. Data loss may occur.

Example:

```
double x = 12.7;
int y = (int) x;
```

After conversion, the decimal part is removed. So the value becomes 12.

3. Type Conversion using Wrapper Classes

Wrapper classes help convert String values into primitive data types. This is very common when taking user input.

Example (String to int):

```
String s = "100";
int n = Integer.parseInt(s);
```

Example (String to double):

```
String d = "45.5";
double v = Double.parseDouble(d);
```

4. Important Points

- Widening conversion is automatic and safe.
- Narrowing conversion needs casting and may cause data loss.
- Wrapper classes are useful for converting String input.
- Wrong String format can cause NumberFormatException.

Conclusion

Type conversion is a basic but very important concept in Java. Once you understand widening, narrowing, and wrapper-based conversion, you can handle data safely in real programs.