📘 Type Casting in Java – Complete Theory Notes

1. What is Type Casting?

➡ **Type Casting** in Java is the process of converting one data type into another.

Java supports this because different data types occupy different memory sizes.

Example:
Convert `int` → `double`
Convert `double` → `int`

2. Two Types of Type Casting
Java supports 2 types of type casting:

✔ 1. Widening (Automatic Type Casting)

➡ Happens automatically
➡ Smaller data type → Larger data type
➡ No data loss
➡ Also called Implicit Casting / Upcasting

byte → short → int → long → float → double

✔ 2. Narrowing (Manual Type Casting)

➡Performed **manually**
➡ Larger data type → Smaller data type
➡ Possible data loss
➡ Also called **Explicit Casting / Downcasting**

double → float → long → int → short → byte

Why Widening is Safe?

- **No data is lost**
- **Smaller data fits safely into a larger container**
- **Java handles the conversion internally**

Why Narrowing is Risky?

- Data might be **lost**
- Overflow may occur
- Manual type casting is required

Type Casting Between char and int

```
char c = 'A';
int ascii = c;
```

Type Casting Between String and Numbers

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

Important Notes & Rules

✔ Widening is automatic
✔ Narrowing must be done manually
✔ Narrowing may cause data loss
✔ `char` stores ASCII values internally
✔ `String` cannot be directly cast → must use wrapper classes
✔ Boolean cannot be type cast to any type