**Casting primitives in java**

casting is the process of converting a variable from one type to another. There are two types of casting in Java:

1. **Widening Casting (Implicit):**
   * This is automatically done by the Java compiler.
   * It occurs when converting a smaller type to a larger type size.
   * Example: byte to short, short to int, int to long, long to float, float to double.
2. **Narrowing Casting (Explicit):**
   * This has to be done manually by the programmer.
   * It occurs when converting a larger type to a smaller size type.
   * Example: double to float, float to long, long to int, int to short, short to byte.

**Example of Widening Casting (Implicit)**

public class WideningCastingExample {

public static void main(String[] args) {

int myInt = 9;

double myDouble = myInt; // Automatic casting: int to double

System.out.println("Integer value: " + myInt);

System.out.println("Double value: " + myDouble);

}

}

In this example, myInt is automatically cast to a double without any explicit cast syntax.

**Example of Narrowing Casting (Explicit)**

public class NarrowingCastingExample {

public static void main(String[] args) {

double myDouble = 9.78;

int myInt = (int) myDouble; // Manual casting: double to int

System.out.println("Double value: " + myDouble);

System.out.println("Integer value: " + myInt);

}

}

In this example, myDouble is manually cast to an int using the cast syntax (int), which truncates the decimal part.

**Key Points to Remember:**

* **Widening Casting** is safer as there is no data loss.
* **Narrowing Casting** can lead to data loss, so it needs to be done carefully.
* When narrowing, the cast is explicitly specified by the programmer to indicate that they are aware of the potential data loss and still want to proceed.