

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

        byte someByte;
        int someInt;
        long someLong;
        float someFloat;
        double someDouble;

        //-----
        //Part I
        //Syntax for casting primitive types
        //-----

        //widening can be done implicitly
        someByte = 37;
        someInt = someByte;
        someFloat = someInt;

        //narrowing must be explicit (with cast)
        someInt = 37;
        // someByte = someInt;    compile-time error, cast needed
        someByte = (byte)someInt;    //correct

        //-----
        //Part II
        //Casting may result in loss of information
        //-----
        System.out.println("Casting may lose information..." );

        System.out.println("Narrowing:" );

        //ok: small int --> byte
        someByte = (byte)127;
        System.out.println("(byte)127 == " + someByte);

        //trouble: large int --> byte
        //High order bits are discarded, so result
        //may change magnitude and sign as well
        someByte = (byte)255;
        System.out.println("(byte)255 == " + someByte);

        //trouble: float-point-->integer
        //Decimals are lost
        someInt = (int)12.8f;
        System.out.println("(int)12.8f == " + someInt);

        //trouble: even when a floating point number appears
        //not to have decimals, they might be lurking
        someDouble = 4.56;
        someDouble = someDouble * 100;
        someLong = (long)(someDouble);
        System.out.println("4.56 * 100 == " + someLong); //prints 455
        //Explanation:
        //someDouble is actually 455.999999999999609201495...
        //so someLong gets 455 (trimmed decimals), not 456

        System.out.println("Widening:" );

        //ok: int --> long
        someLong = (long)5000034;
        System.out.println("(long)5000034 == " + someLong);
    }
}
```

```
//trouble: long --> float  
//A floating point number has fewer significant  
//digits than a long (7 vs 19)  
someInt = 1234567890;  
someFloat = someInt;  
System.out.println("1234567890 - 1234567890 == "  
                    + (someInt - (int)someFloat));  
}  
}
```