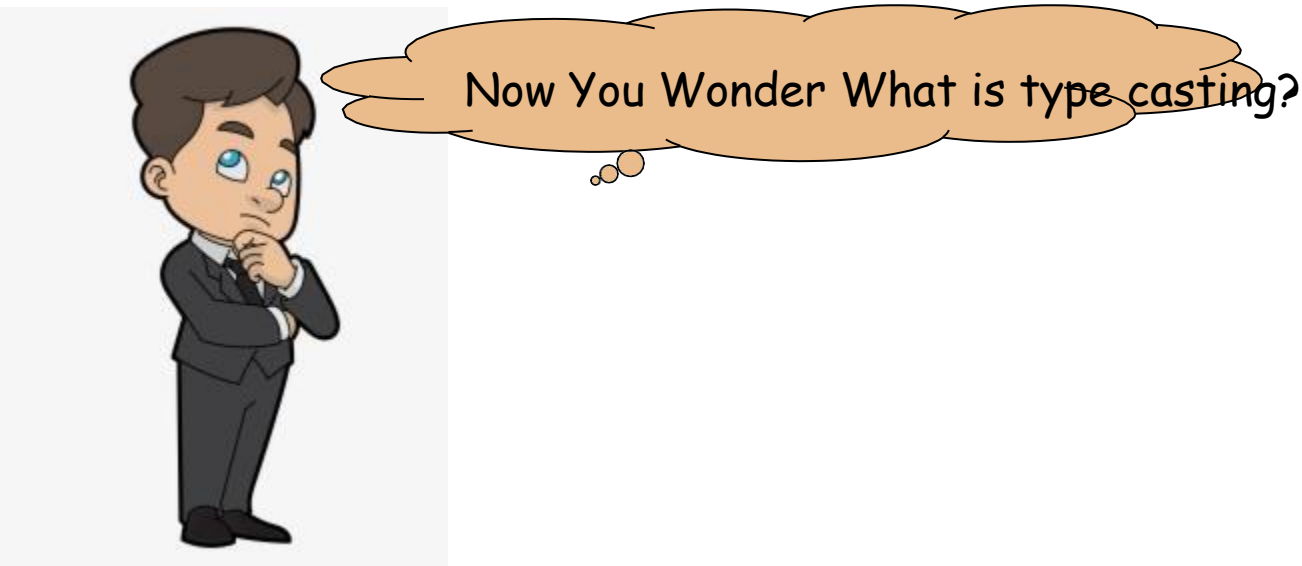
Type casting in java:



Type casting is a process of converting one type of data to another

In Java, there are two types of casting:

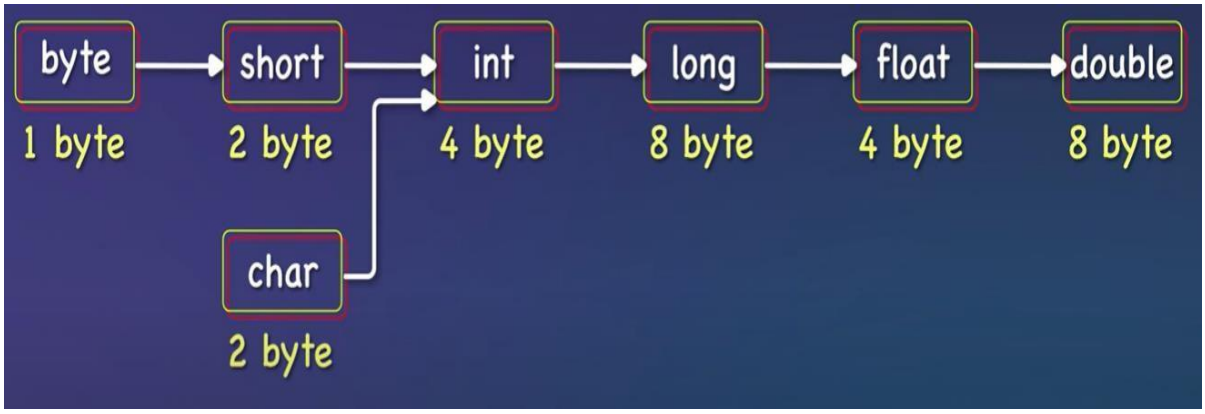
Implicit casting (automatically) - converting a smaller type to a larger type size
byte -> short -> char -> int -> long -> float -> double
Explicit casting (manually) - converting a larger type to a smaller size type

Implicit type Casting:

When a smaller data type is converted to a larger data type, the conversion is automatically performed by **the java compiler** and is referred to as implicit type casting.

Advantage: No loss of precision

Consider the **Implicit type casting** chart given below to understand this:



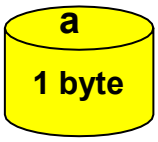
Orders of Implicit Type-Casting for Primitives

Let us consider a code snippet to understand this:

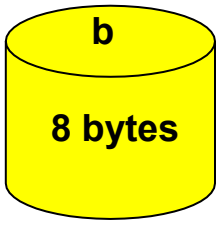
```
byte a = 45;
double b;
b =a;
```

Let us understand implicit type casting using the above code snippet

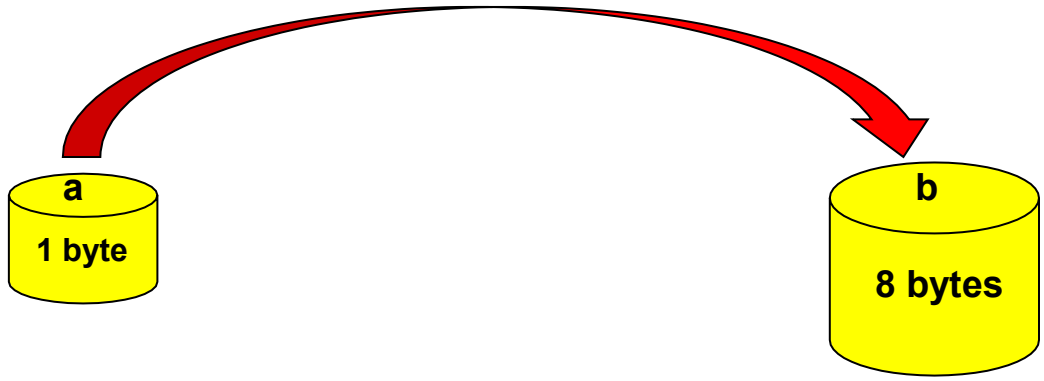
a is a variable of type byte whose size is 1 byte



b is a variable of type byte whose size is 8 byte



b = a; we are now trying to store the data present in a into b;
a is of type byte and can store 1 byte. b is of type double and can store 8 bytes. We are trying to store data of smaller size into larger size.



This conversion is implicitly done without user interaction and hence it is referred to as implicit type casting

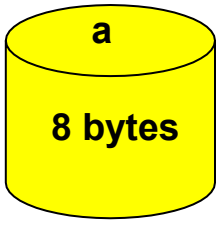
Explicit type Casting:

When a larger data type is converted to a smaller data type, the **conversion is not automatically performed** by **the java compiler** and must be done by **programmer explicitly** and hence it is referred to as explicit type casting

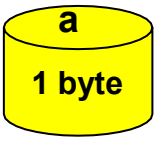
Let us consider a simple code snippet to understand this, the way we understood explicit type casting

```
double a = 45.5;
byte b;
b = a;
```

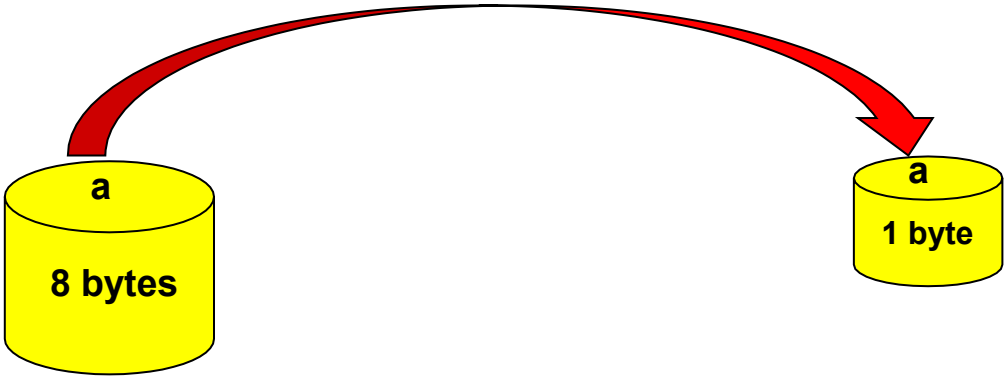
a is a variable of type double whose size is 8 bytes.



b is a variable of type byte whose size is 1 byte.



b=a; will give you **error** as you are trying to store a larger type of data into a smaller type.



The above conversion will result in error as loss of precision occurs. To get the error free output, we have to explicitly convert the data as shown below

```
double a = 45.5;
byte b;
b = (byte)a;
```

b is of type byte and it will only store 45 and 0.5 is lost during the conversion which is the disadvantage of explicit type casting.