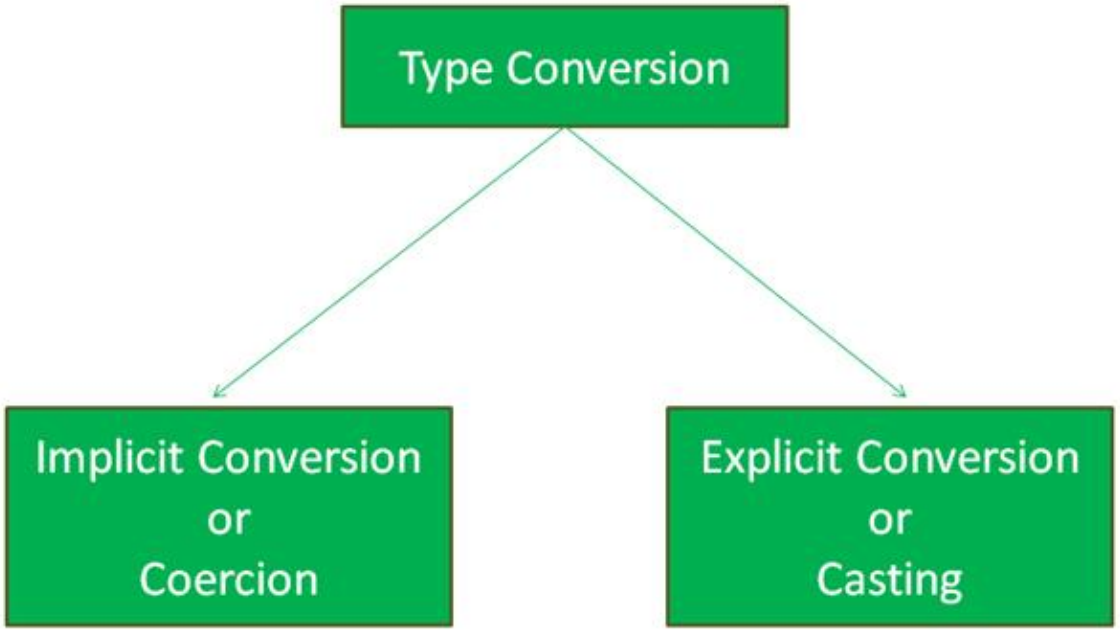
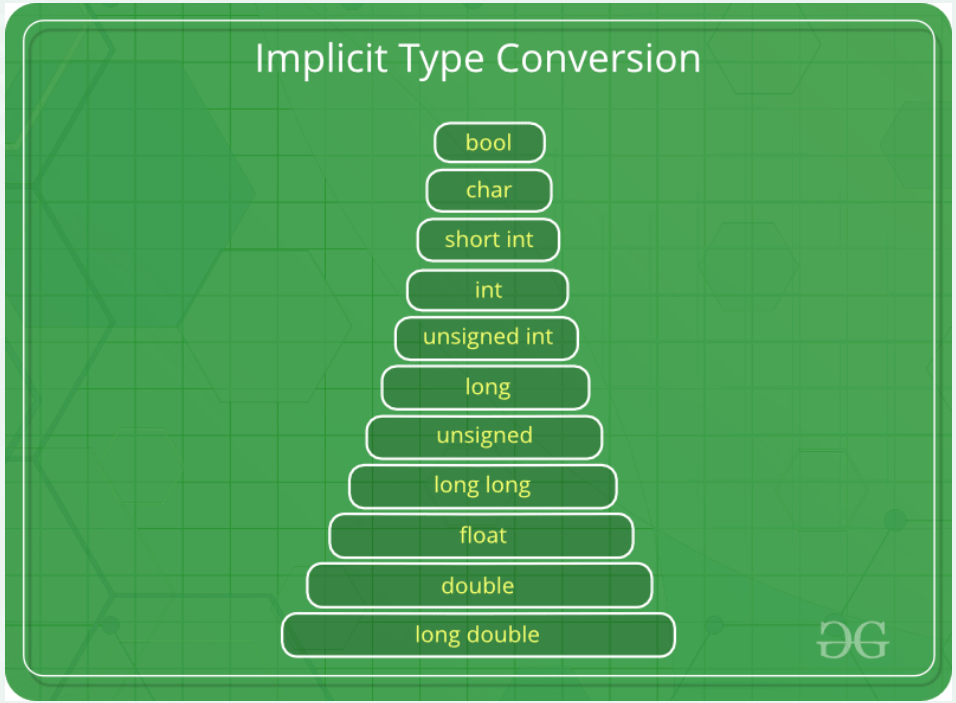
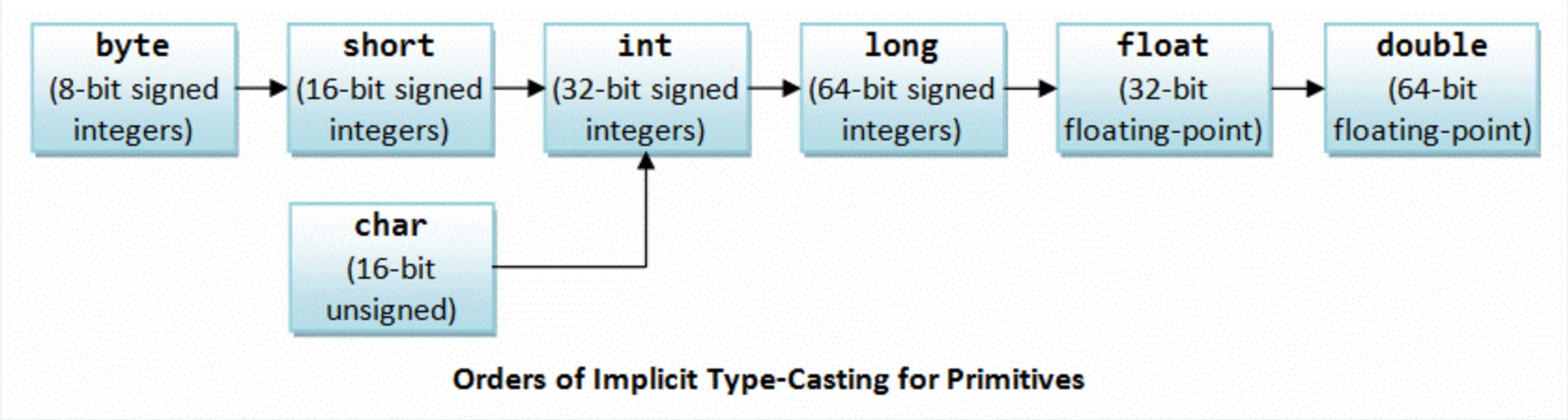
Ma’lumotlarni bir turdan boshqa turga o`tkazish (Type Conversion)


Type Conversion
Implicit Conversion or Coercion
Explicit Conversion or Casting
Startertutorials.com

- Kengaytirish yoki Avtomatik konvertatsiya (Widening or Automatic Type Conversion)
- Qisqartirish yoki aniq konvertatsiya (Narrowing or Explicit Conversion)


Implicit Type Conversion
bool
char
short int
int
unsigned int
long
unsigned
long long
float
double
long double

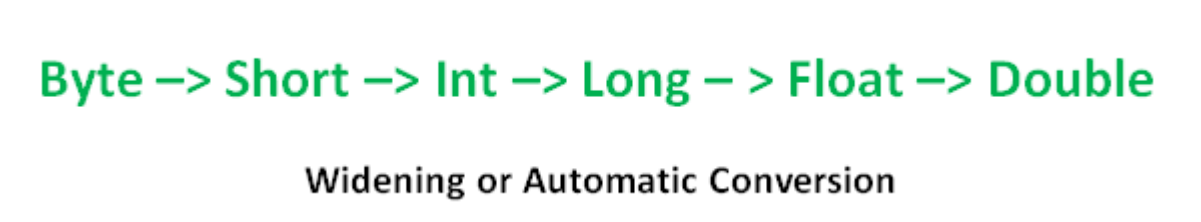


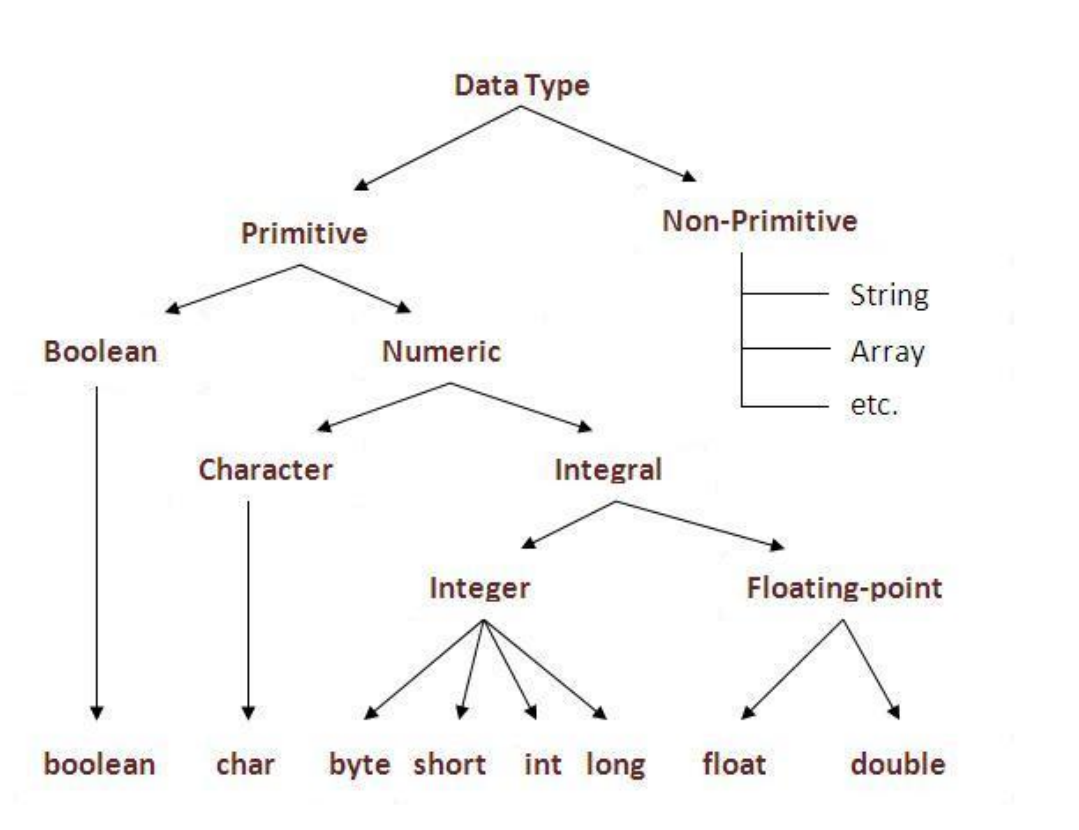
Orders of Implicit Type-Casting for Primitives

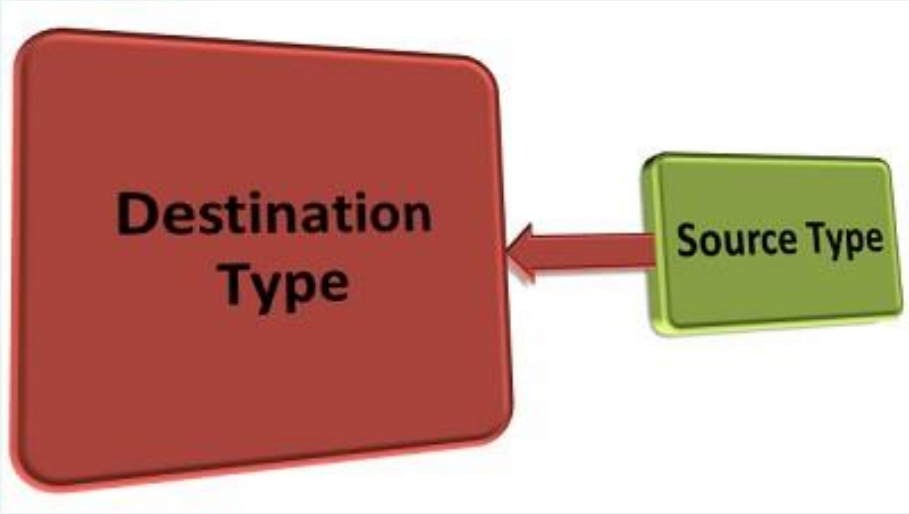
Widening or Automatic Type Conversion

Avtomatik konvertatsiya quyidagi shartlar bajarilganda amalga oshadi

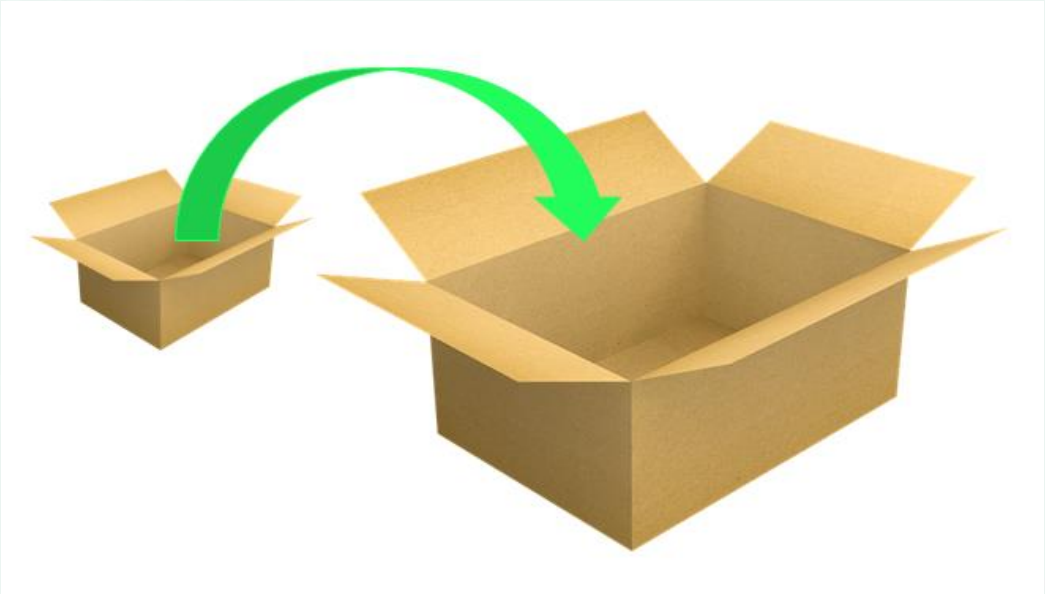
- The two data types are compatible. (Ikkala ma’lumot toifasi bir-biriga mos kelganda)
- When we assign value of a smaller data type to a bigger data type. (Kichik hajmli ma’lumot toifasini katta hajmli ma’lumot toifasiga o’zlashtirganda)



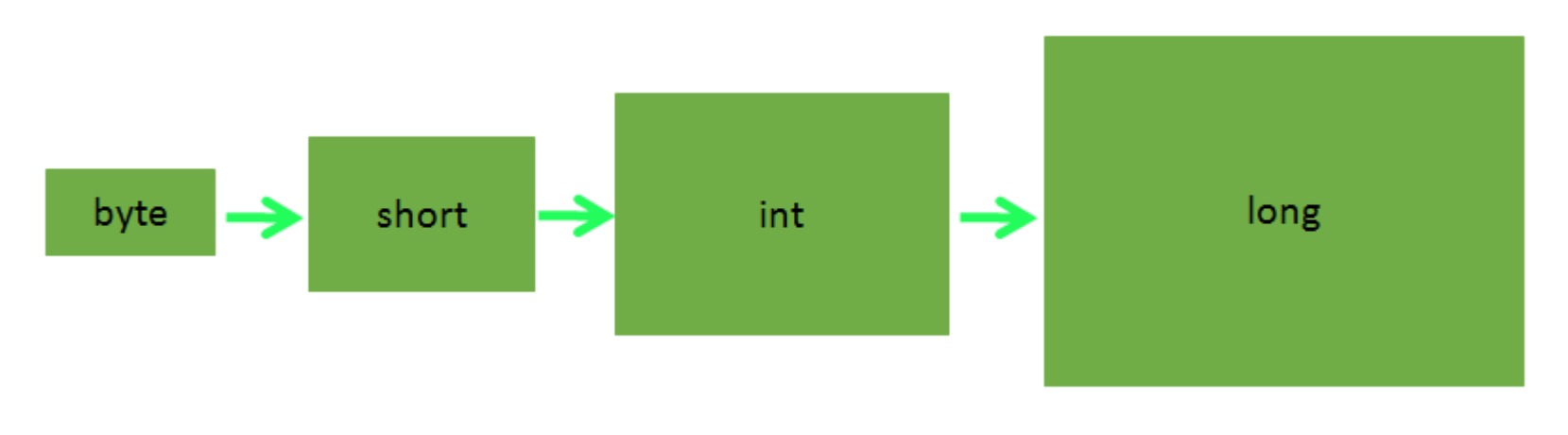

Data Type
Primitive
Non-Primitive
String
Array
etc.
Boolean
Numeric
Character
Integral
Integer
Floating-point
boolean
char
byte
short
int
long
float
double

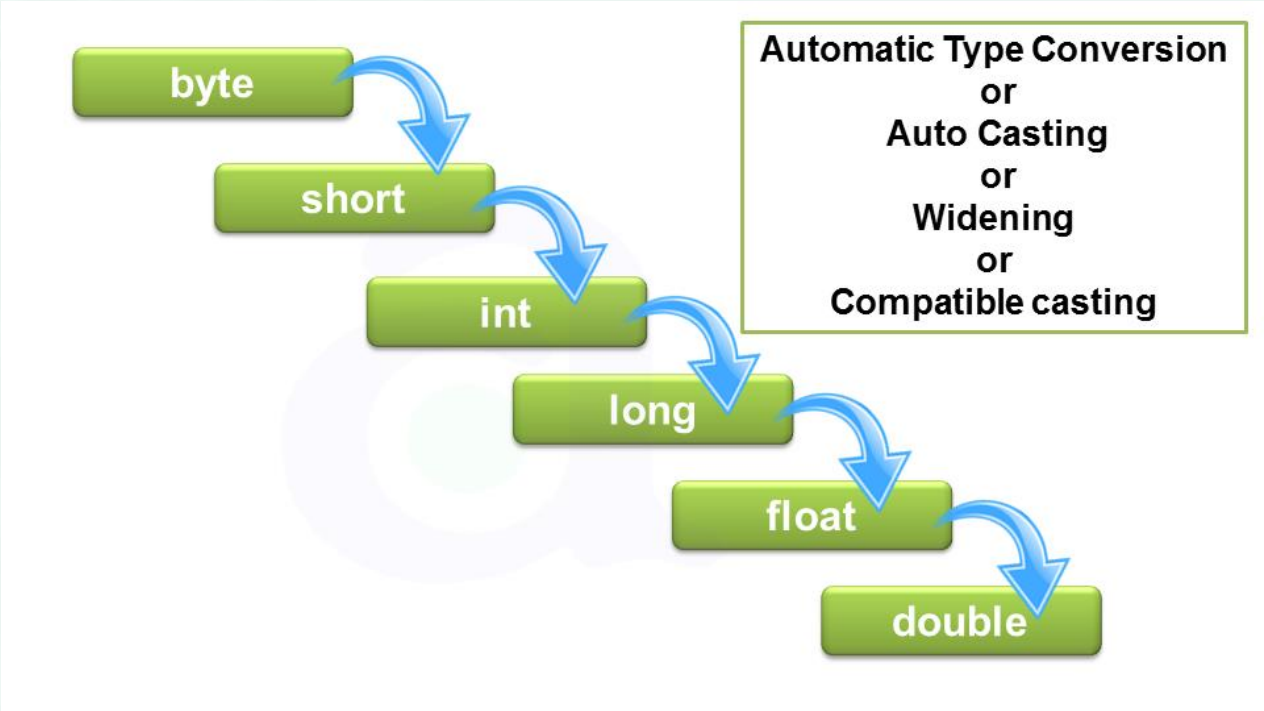

Destination Type
Source Type

Widening or Automatic Type Conversion



Widening or Automatic Type Conversion




byte
short
int
long
float
double
Automatic Type Conversion
or
Auto Casting
or
Widening
or
Compatible casting

- Widening Type Conversion
  - Implicit conversion by compiler automatically

byte -> short, int, long, float, double
short -> int, long, float, double
char -> int, long, float, double
int -> long, float, double
long -> float, double
float -> double

byte to byte

```
class Test {
        public static void main(String[] args) {
                byte a = 15;
                byte b = a;
                System.out.println(b);
        }
}
```

15

byte to int

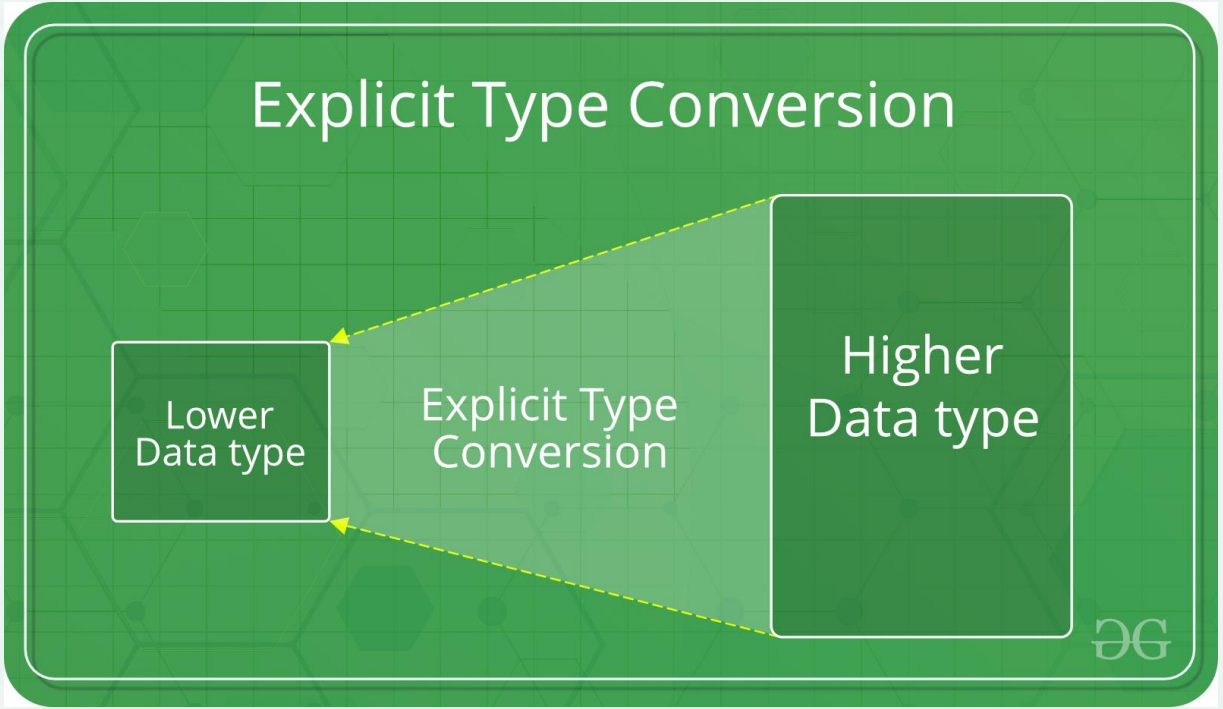
```
class Test {
        public static void main(String[] args) {
                byte a = 15;
                int b = a;
                System.out.println(b);
        }
}
```

15

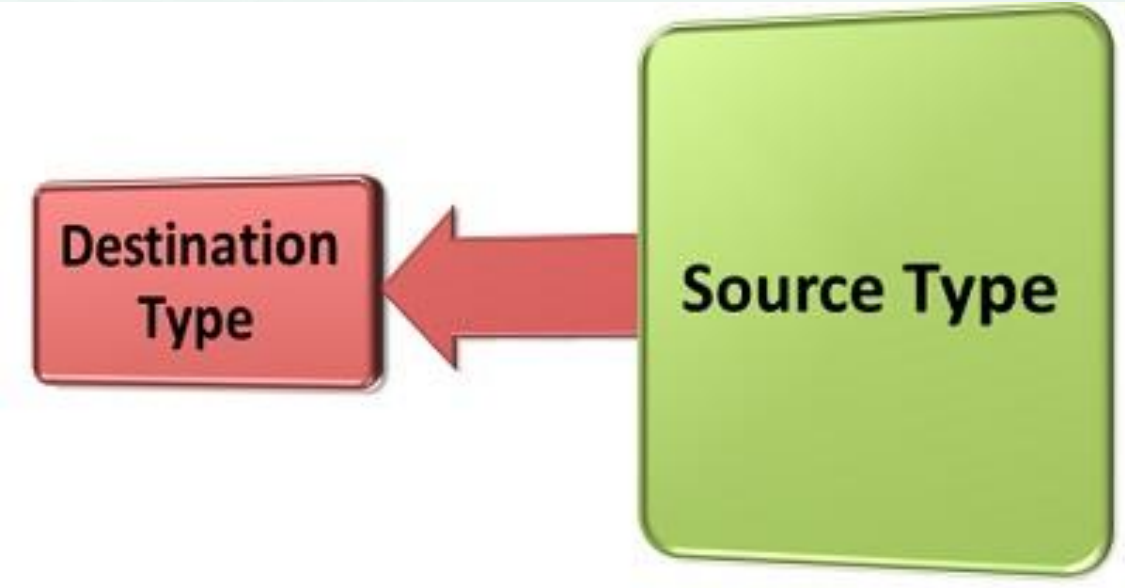
Widening or Automatic Type Conversion

```
class Test {
    public static void main(String[] args)    {
        int i = 100;
        //automatic type conversion
        long l = i;
        //automatic type conversion
        float f = l;
   }
}
```

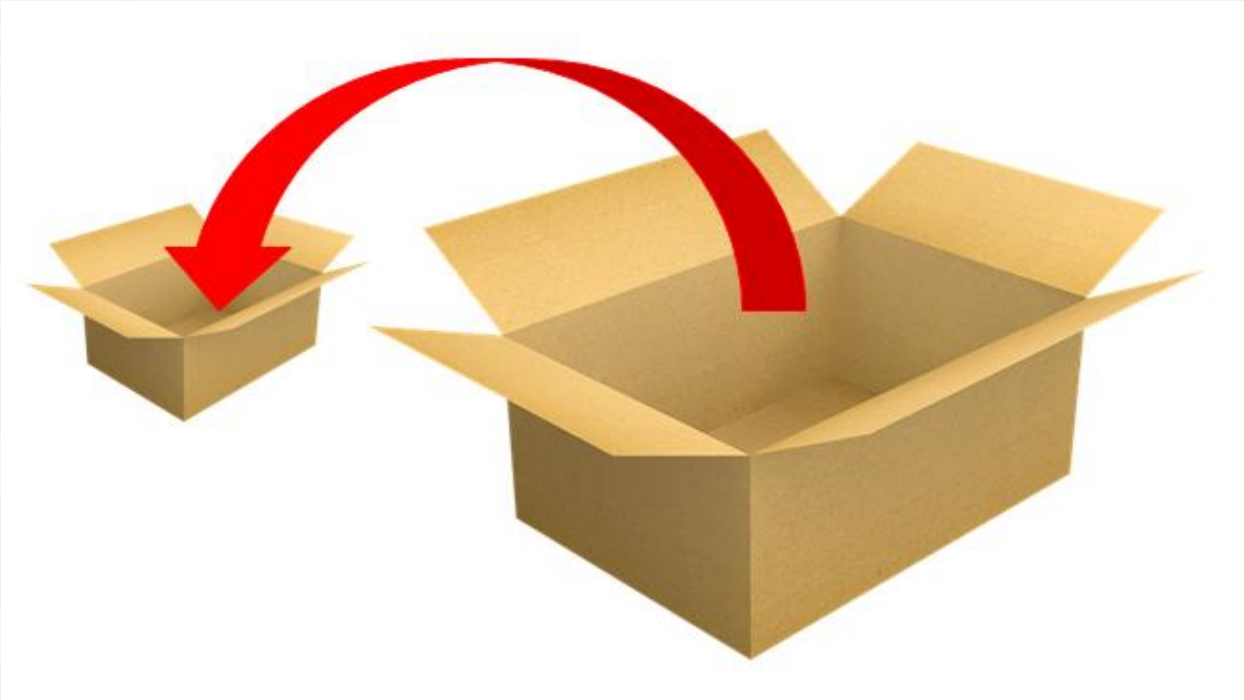
Narrowing or Explicit Conversion



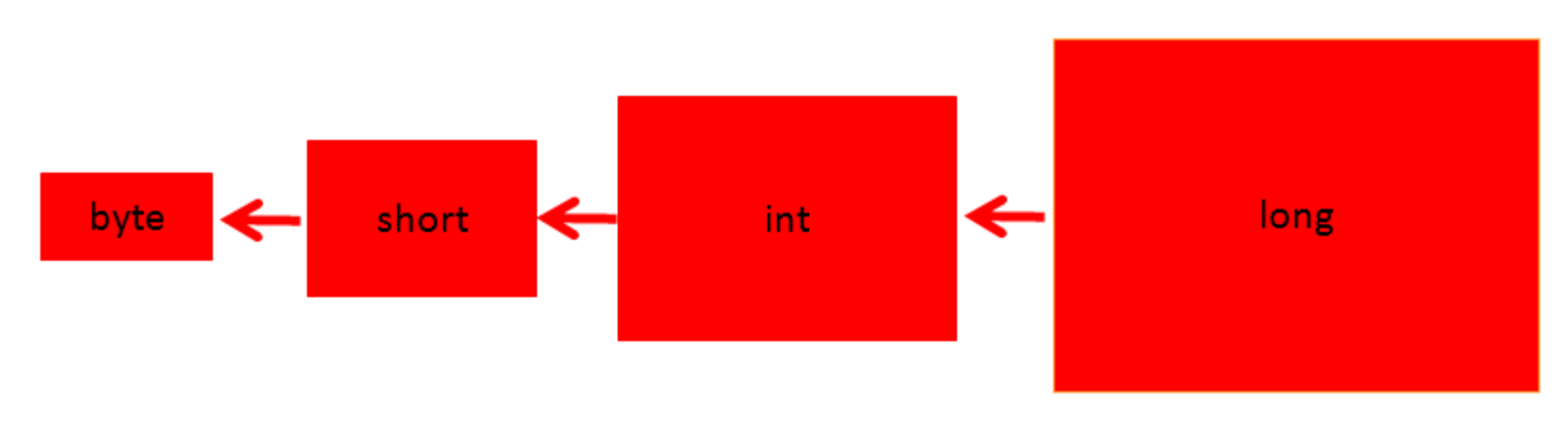
Narrowing or Explicit Conversion

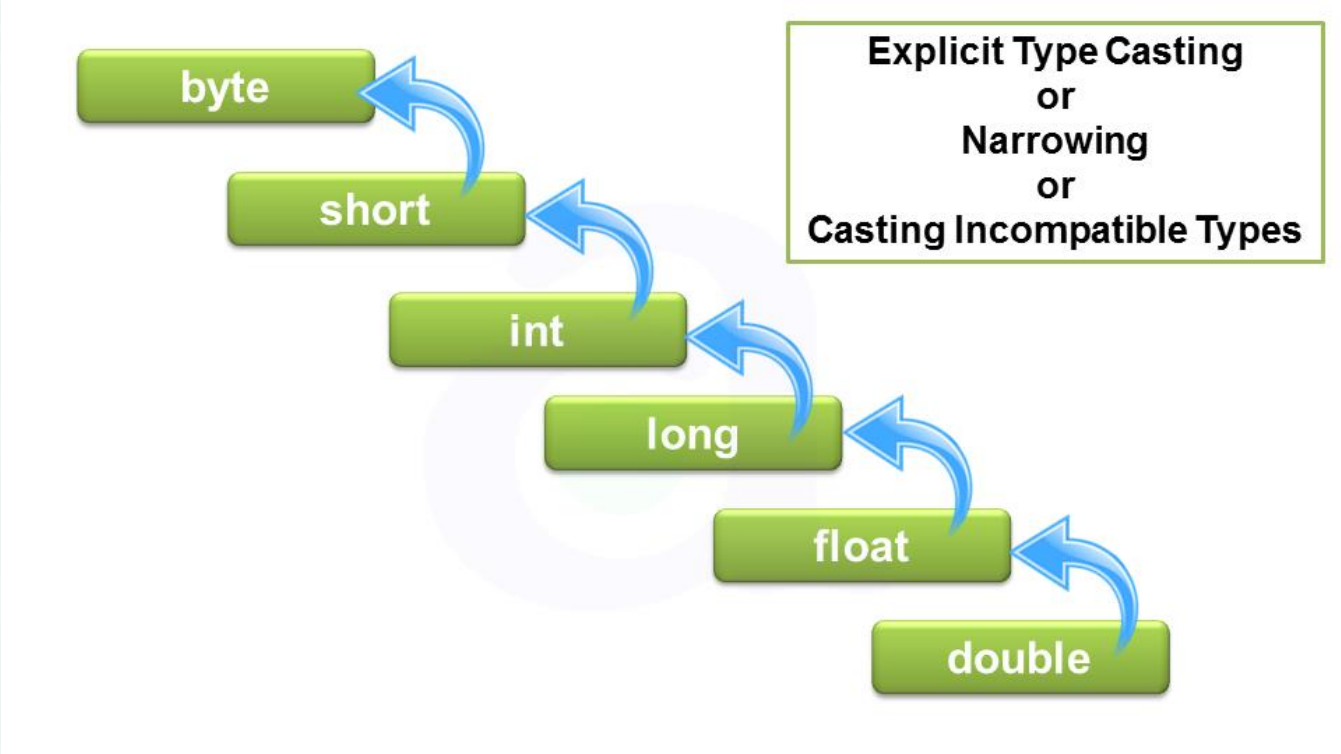


Narrowing or Explicit Conversion



Narrowing or Explicit Conversion




byte
short
int
long
float
double
Explicit Type Casting
or
Narrowing
or
Casting Incompatible Types

byte -> char

short -> byte, char

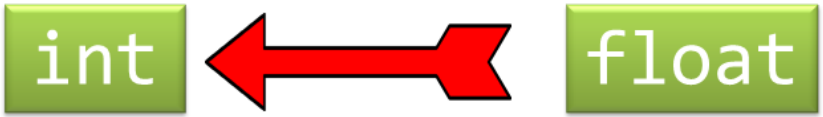
char -> byte, short

int -> byte, short, char

long -> byte, short, char, int

float -> byte, short, char, int, long

double -> byte, short, char, int, long, float




```
int number;
float fval= 32.33f;
number= (int)fval;
```

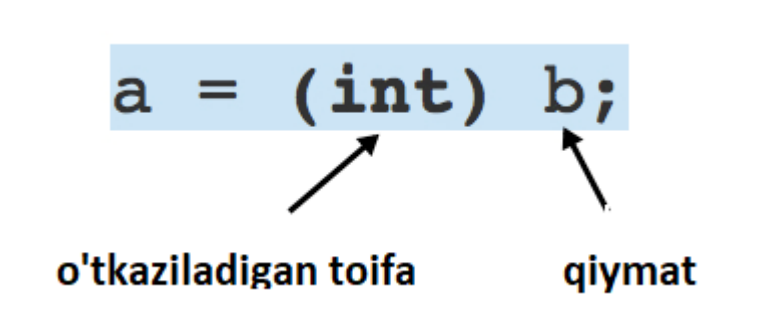
Type in which you want to convert

Variable name
Which you want to convert

Narrowing or Explicit Conversion

```
class Test {
   public static void main(String[] args) {
            int a=0;
            long b=15;
            a = (int) b;
   }
}
```

Narrowing or Explicit Conversion



Narrowing or Explicit Conversion

```
class Test{
    public static void main(String[] args){
        double d = 100.04;
                    //explicit type casting
        long l = (long)d;
                    //explicit type casting
        int i = (int)l;
    }
}
```

Double –> Float –> Long –> Int –> Short –> Byte

Narrowing or Explicit Conversion

Narrowing or Explicit Conversion

```
class Test{
   public static void main(String args[]){
      byte b;
      int i = 257;
      double d = 323.142;
         //i%256
      b = (byte) i;
      System.out.println("i = " + i + " b = " + b);
         //d%256
      b = (byte) d;
      System.out.println("d = " + d + " b= " + b);
   }
}
```

short, char, int, long to byte

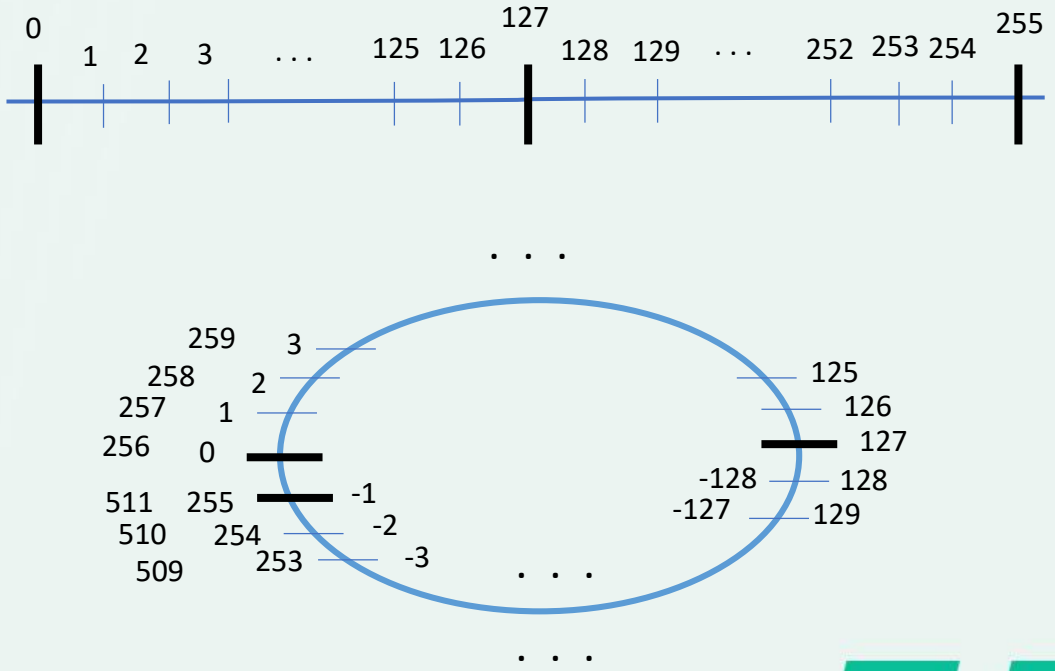
1 bit = 0 yoki 1

1 byte = 8 bit

00000000= 0
00000001= 1
00000010= 2
00000011= 3
00000100= 4

……..

11111011= 251
11111100 = 252
11111101 = 253
11111110 = 254
11111111= 255



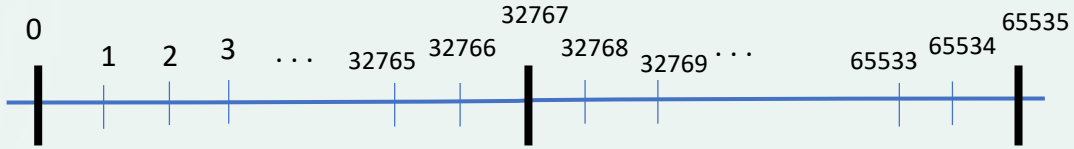
int , long to short

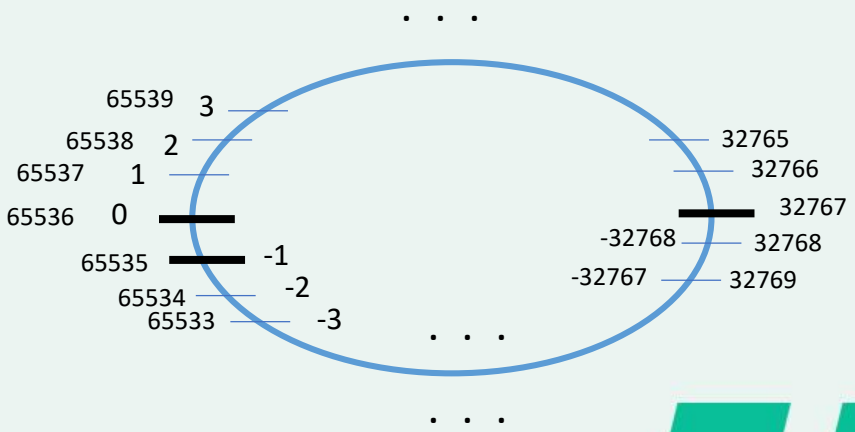
2 byte = 16 bit

00000000 00000000= 0
00000000 00000001= 1
00000000 00000010= 2
00000000 00000011= 3
00000000 00000100= 4

……..

11111111 11111011= 65531
11111111 11111100 = 65532
11111111 11111101 = 65533
11111111 11111110 = 65634
11111111 11111111 = 65535

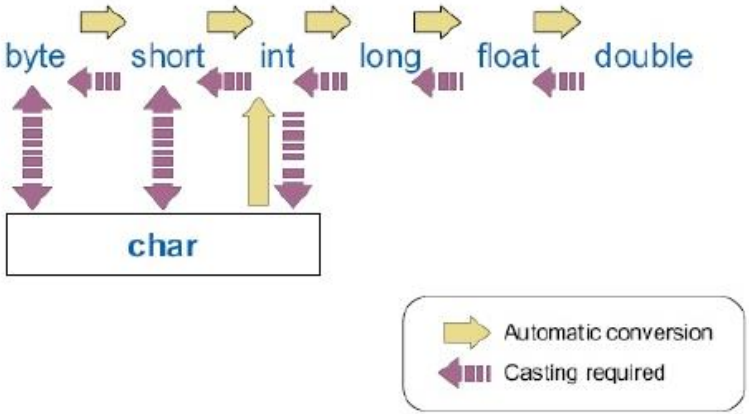




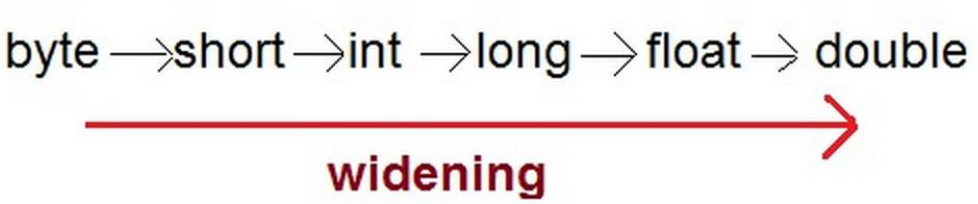
PERSONAL
DEVELOPMENT PROCESS
stay weird, stay different

Implicit versus Explicit Casting

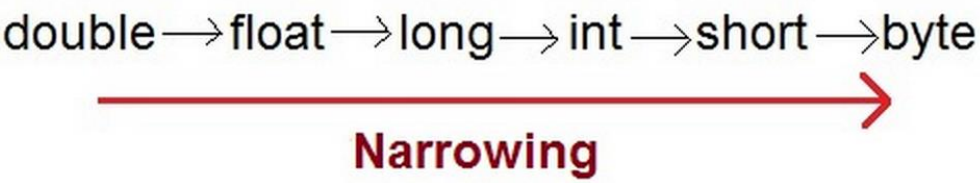
- Implicit casting is automatic when no loss of information is possible
- An explicit cast required when there is a “potential” loss of accuracy



- Widening Casting(Implicit)

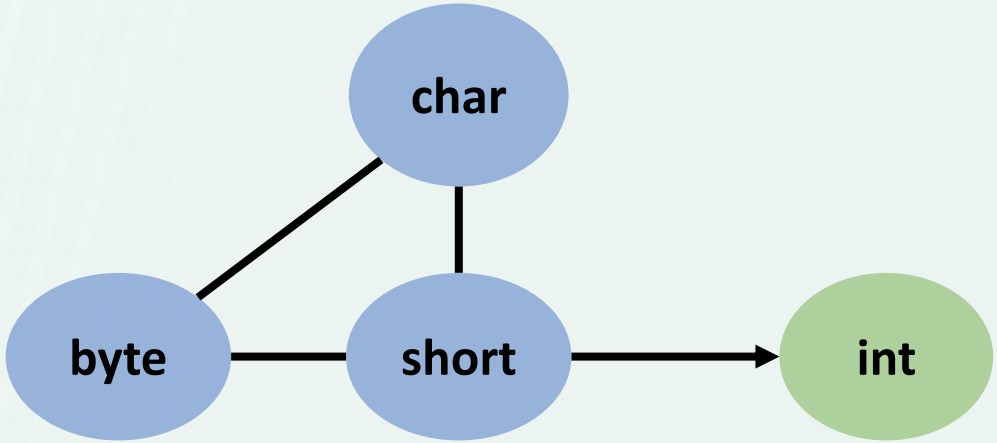


- Narrowing Casting(Explicitly done)

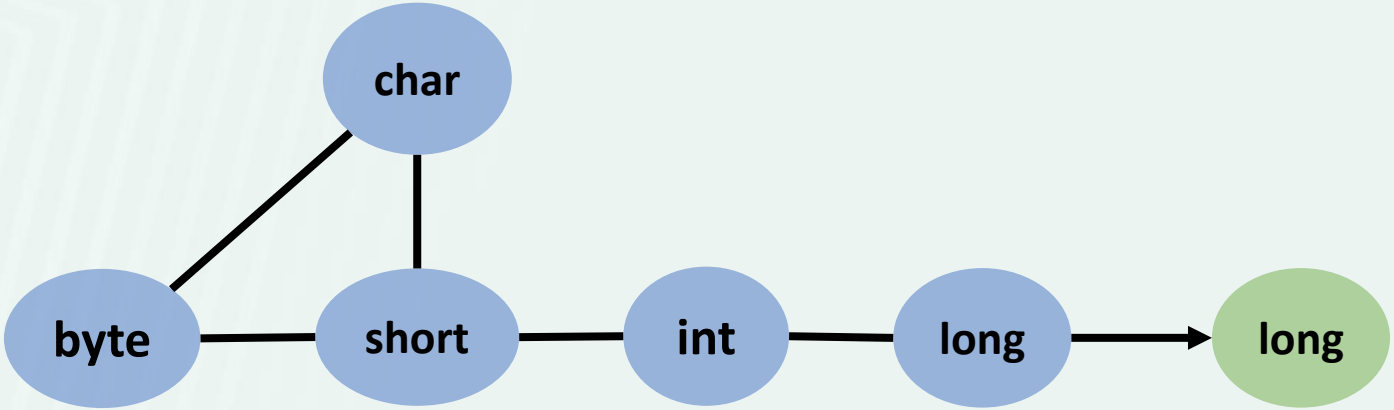


Arifmetik ifoda orqali ma’lumotlar toifasini o’zgartirish

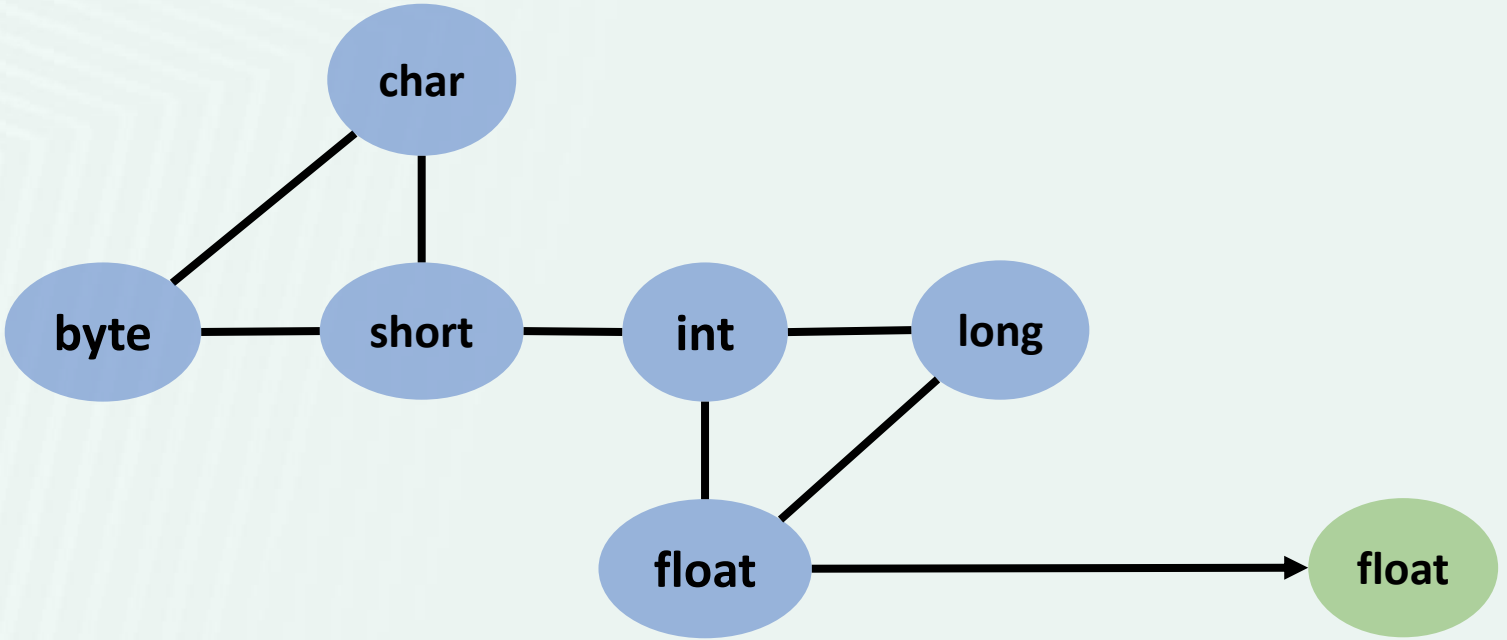
byte, short va ***char*** toifalar ishtirok etgan arifmetik ifoda natijasining toifasi ***int*** bo'ladi. Ya'ni ushbu uchala toifa ham avtomatik tarzda ***int*** *ga o'giriladi*



Agarda arifmetik ifodada bitta ***long*** toifali o’zgaruvchi ishtirok etgan bo’lsa ifoda qiymatining toifasi ***long*** bo’ladi.



Agarda arifmetik ifodada bitta ***o'zgaruvchan vergulli*** toifali o'zgaruvchi ishtirok etgan bo'lsa ifoda qiymatining toifasi ***o'zgaruvchan vergulli*** bo'ladi.



Agarda arifmetik ifodada bitta ***double*** toifali o’zgaruvchi ishtirok etgan bo’lsa ifoda qiymatining toifasi ***double*** bo’ladi.