

## Type Conversion in Compiler Design

- Conversion of one data type to another automatically by the compiler is called "Type Conversion".
- For Example, Consider expressions like  $x + i$ , where  $x$  is of type float and  $i$  is of type integer.
- Since the representation of integers and floating-point numbers is different within a computer and different machine instructions are used for operations on integers and floats, the compiler may need to convert one of the operands of  $+$  to ensure that both operands are of the same type when the addition occurs.
- Suppose that integers are converted to floats when necessary, using a unary operator (float).

For example, the integer 2 is converted to a float in the code for the expression  $2 * 3.14$ :

**t1 = (float) 2**

**t2 = t1 \* 3.14**

- The attribute  $E.type$ , whose value is either integer or float.
- The rule associated with  $E \rightarrow E1 + E2$  builds on the pseudocode



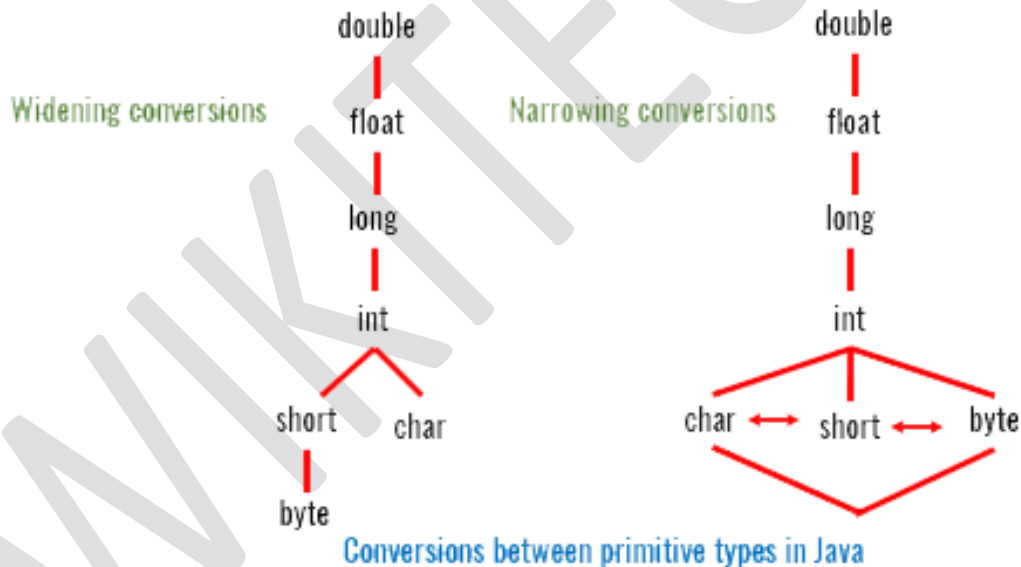
if (E1.type = integer and E2.type = integer) E.type = integer;

else if (E1.type = float and E2. type = integer) E.type = float;

else if (E1.type = integer and E2. type = float) E.type = float;

else if (E1.type = float and E2. type = float) E.type = float;

- Type conversion rules vary from language to language. The rules for Java in the distinguish between widening conversions, which are intended to preserve information, and narrowing conversions, which can lose information.



**For More Details Click Here:**

<https://www.wikitechy.com/tutorials/compiler-design/type-conversion-in-compiler-design>

