The Type Promotion Rules

Widening conversions do not lose information about the magnitude of a value.

For example, an `int` value is assigned to a `double` variable.

This conversion is legal because doubles are wider than ints.

Java's widening conversions are

- From a byte to a short, an int, a long, a float, or a double
- From a short to an int, a long, a float, or a double
- From a char to an int, a long, a float, or a double
- From an int to a long, a float, or a double
- From a long to a float or a double
- From a float to a double

Widening conversions:

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

Here are the Type Promotion Rules:

1. All `byte` and `short` values are promoted to `int`.
2. If one operand is a `long`, the whole expression is promoted to `long`.
3. If one operand is a `float`, the entire expression is promoted to `float`.
4. If any of the operands is `double`, the result is `double`.

In the following code, `f * b`, `b` is promoted to a `float` and the result of the subexpression is `float`.

```
public class Main {

  public static void main(String args[]) {
    byte b = 4;
    float f = 5.5f;
    float result = (f * b);
    System.out.println("f * b = " + result);
  }
}
```

The output:

```
f * b = 22.0
```

In the following program, `c` is promoted to `int`, and the result is of type `int`.

```
public class Main {
  public static void main(String args[]) {
    char c = 'a';
    int i = 50000;
    int result = i / c;
    System.out.println("i / c is " + result);
  }
}
```

The output:

```
i / c is 515
```

In the following code the value of `s` is promoted to `double`, and the type of the subexpression is `double`.

```java
public class Main {
  public static void main(String args[]) {
    short s = 1024;
    double d = .1234;
    double result = d * s;
    System.out.println("d * s is " + result);
  }
}
```

The output:

```
d * s is 126.3616
```

« Automatic Type Promotion in Expressions

Operators »