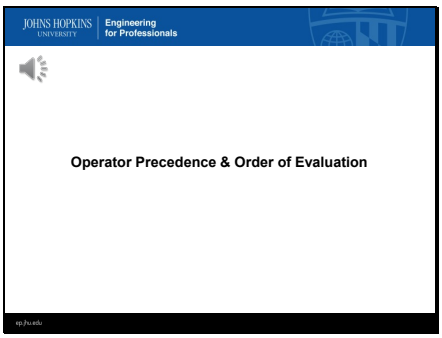
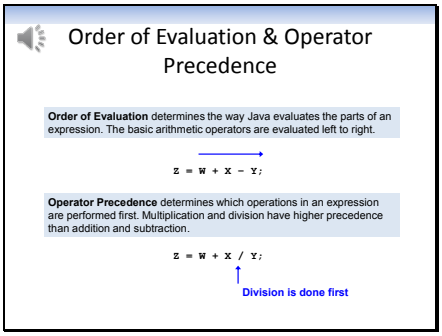
1



In this lecture you will learn about arithmetic operator precedence and order of evaluation.

2



There are two important things you need to be aware of when you use the basic arithmetic operators in expressions…ORDER OF EVALUATION and OPERATOR PRECEDENCE.

The ORDER OF EVALUATION of an operator determines the way Java evaluates the parts of an expression the operator is used in. The order of evaluation for the 4 basic arithmetic operators is LEFT to RIGHT.

Consider the statement Z = W + X – Y. When executing this statement, Java starts with the right-hand side expression. It takes the current value of W, then adds the value of X, then subtracts the value of Y. It performs these steps left to right.

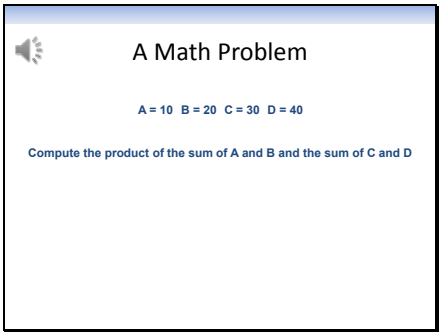
OPERATOR PRECEDENCE determines which operations in an expression are performed first. Some Java operators have higher precedence than others, and this can change the order in which parts of an expression are evaluated. Multiplication and Division have higher precedence than addition and subtraction.

If you look at the earlier example, you see that there is an addition operation and a subtraction operation. Since addition and subtraction have the same operator precedence the expression was evaluated left to right.

Now, let’s take the following example. This statement contains an addition operation and a division

operation. Since division has higher precedence that addition, the division calculation will be performed first, and then the right-hand side will be evaluated left to right.

3



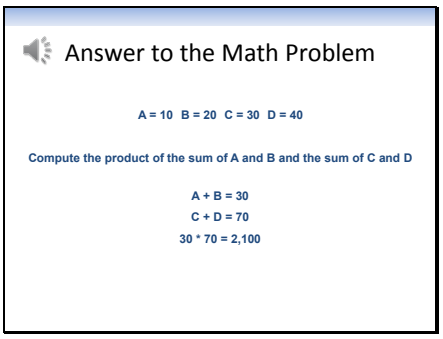
It's very important that you understand operator precedence and order of evaluation so that you can avoid unwanted calculation errors in your programs.

I'm going to stop talking about programming for just a few minutes, and I'm going to ask you to do a math problem.

There are 4 variables…a, b, c, and d. Each variable is assigned a value. The problem is to compute the product of the sum of a and b and the sum of c and d.

I'd like you to solve this problem for me. Go ahead and pause this lecture while you do the calculations…then, continue the playback to see the answer.

4

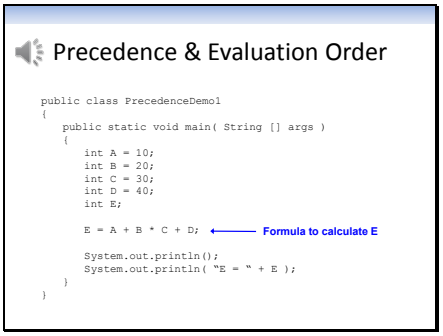


Okay…let's see how my solution compares with yours.

The sum of A & B is 30. The sum of C & D is 70. So the product of 30 and 70 is 2,100.

Now, let's write a program to calculate this.

5



Look at this sample program. Four integer variables are declared, and a simple calculation is made and displayed. What I want this program to do is to take the sum of a & b and multiply it by the sum of c & d, like in the math problem you just completed. In this program I've declared another variable, named E to store the result.

I'd like you pause this lecture now.

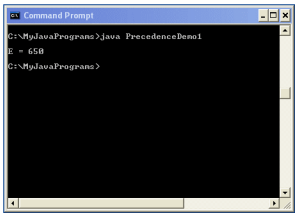
Next, I'd like you to look at the formula used to calculate a value for the variable E.

Now, using the values assigned to A, B, C, and D, I'd like you to calculate how this formula will be evaluated.

When you're done, continue the lecture and we'll run the program and see what the answer is.

6

Precedence & Evaluation Order



If your answer was 650 you've evaluated the expression in the program correctly…at least that's what the program will produce. But that's not the correct answer to our math problem. What I really wanted the program to do is to sum A and B, sum C and D, and then compute the product of these two sums.

Let's see what happened.

7

Precedence & Evaluation Order

```
public class PrecedenceDemo1
{
    public static void main( String [] args )
    {
        int A = 10;
        int B = 20;
        int C = 30;
        int D = 40;
        int E;

        E = A + B * C + D;

        System.out.println();
        System.out.println( "E = " + E );
    }
}
```

This program's output will be 650, but the correct answer to the math problem we wanted to solve is 2,100.

The problem here is that the calculation expression is not doing what it needs to do…which is to calculate the product of the sum of A & B and the sum of C & D.

Here's how the computer will make its calculation. First, it will take the product of B and C, because multiplication has higher precedence than addition. That gives an intermediate result of 600. Then, it will add the value of A to the intermediate result, making it 610, and then it will add the value of D, resulting in 650.

What I really needed the program to do is to first take the sums of A & B and C & D, then multiply the resulting sums. So, how can I get the program to do the correct calculations?

8

Parentheses Impact Evaluation Order

```
public class PrecedenceDemo2
{
    public static void main( String [] args )
    {
        int A = 10;
        int B = 20;
        int C = 30;
        int D = 40;
        int E;

        E = (A + B) * (C + D);

        System.out.println();
        System.out.println( "E = " + E );
    }
}
```

I need to get the program to do the calculation operations in the correct order.

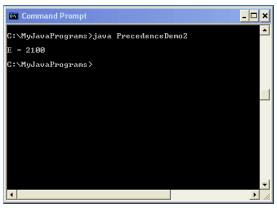
This can be done by having it first compute the sums and then do the multiplication operation. By placing parentheses around the components of the expression I can have the program calculate in the correct way.

When Java evaluates expressions, it will evaluate expressions that are in parentheses first. So, in this example, the sum of A & B will be computed first, then the sum of C & D, and then the multiplication will be performed, giving us the correct result of 2,100.

Let's run the program to be sure.

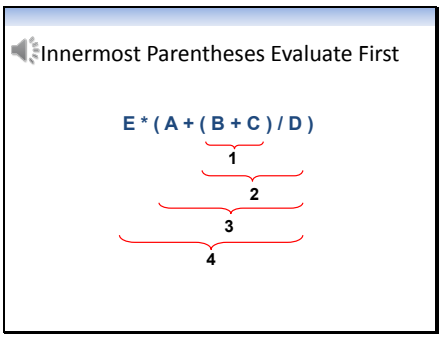
9

Parentheses Impact Evaluation Order



There we go. Now it solves the problem correctly.

10



The concepts of evaluation order and operator precedence are so important they warrant a bit more discussion.

We saw that Java evaluates expression components that are contained in parentheses first, and then applies its rules for precedence and evaluation order to the remainder of the expression.

Sometimes, if we have complicated calculations, there may be several sets of parentheses. The important thing to remember here is that Java evaluates expressions in the innermost set of parentheses first, and works its way out.

Here's an example. In this formula there are two sets of parentheses, and one set is enclosed within, or nested within, the other set. This expression will be evaluated as follows:

First, the expression in the innermost set of parentheses will be evaluated…the sum of B & C.

Then, the expression in the outer set of parentheses is evaluated. Since this expression has an addition and a division operation, the division will be done first [*].

Next, the addition operation will be done [*].

And then, the multiplication will be done [*]. E will be multiplied by the intermediate value.