

Welcome

Getting Started with BlueJ

Variables and Mathematical Operators

Functions and Conditionals

Functions 4 min

Conditionals 3 min

Practice Quiz: Functions and Conditionals 3 questions

Classes, Types, and For Each Loops

Seven Steps for Solving Programming Problems

```
int f (int x, int y) {
    if (x < y) {
        System.out.println("x < y");
        return y + x;
    }
    else {
        System.out.println("x >= y");
        if (x > 8) {
            return y + 7;
        }
    }
    return x - 2;
}

int g () {
    int a = f (3, 4);
    int b = f (a, 5);
    return b;
}
```

g

Output

We have another function g.

Duke UNIVERSITY

Conditionals

Have a question? Discuss this lecture in the week forums.



Interactive Transcript

Search Transcript

English ▾

0:03

Now, let's see how to execute code with conditional statements in it. Here we have a function f, which has some if and if, else statements. [We have another function g.](#) We'll assume we've used BlueJ's interface to invoke the method or function g to start with. The first statement declares a variable a. So we'll make a box for a. Even though this statement initializes a, it's going to take us several steps to compute that value. So we're going to leave a's value as 0, until we compute the value of the method f, applied to arguments 3 and 4. To evaluate this called to f, we make a frame, passing the values of the parameters. Note where to return and then we move the execution arrow to the first line f. The next line of code is an if statement, whose conditional expression is x less than y. We evaluate that expression and find 3 less than 4 is true. So I move the execution arrow into the then clause of this if statement and continue executing. The next statement is a call to System.out.println which is how you print something in Java. We write x less than y in our output and then, we return y plus x. This values follows the same rules you've already learned for return statements. We evaluate the expression to get the return value which is 7. Which is then what the function call evaluates to and will return back to the caller, destroying the frame for f. Now we're ready to finish initializing a, since we know that the call to the method f evaluated to 7. So, a is now 7. Now, we're ready for the next line of code in g. We make a box for b. And we call f passing in the arguments 7 and 5. Once again, we want to evaluate the condition x less than y.

2:04

However, now x is 7 and y is 5. So x less than y is false. We find the closed curly brace for this if statement and see that the if statement has an else clause. So we move the execution arrow into the start of the else clause and continue executing from there. First, we see the System.out.println statement and this prints the line x is greater than or equal to y. Then we reach another if statement. This if statement is nested inside of the else but that doesn't affect the rules of how we evaluate it. We see that the conditional expression is false since 7 is not greater than or equal to 8. There's no else clause, so we move the execution arrow immediately past the closed curly brace and continue execution. There are not anymore statements inside the else clause. So remove the execution arrow outside of the else clause and keep going. Next, we have a return statement. So we evaluate the expression x minus 2 to find that the return value is 5. This then gets returned to the place we noted. So we destroy the frame for f and return back to g. We finished the assignment statement and now we're ready for the return statement from g. We execute that and we're done with method g.

Downloads

Lecture Video mp4

Subtitles (English) WebVTT

Transcript (English) txt

Would you like to [help us translate](#) the transcript and subtitles into additional languages?