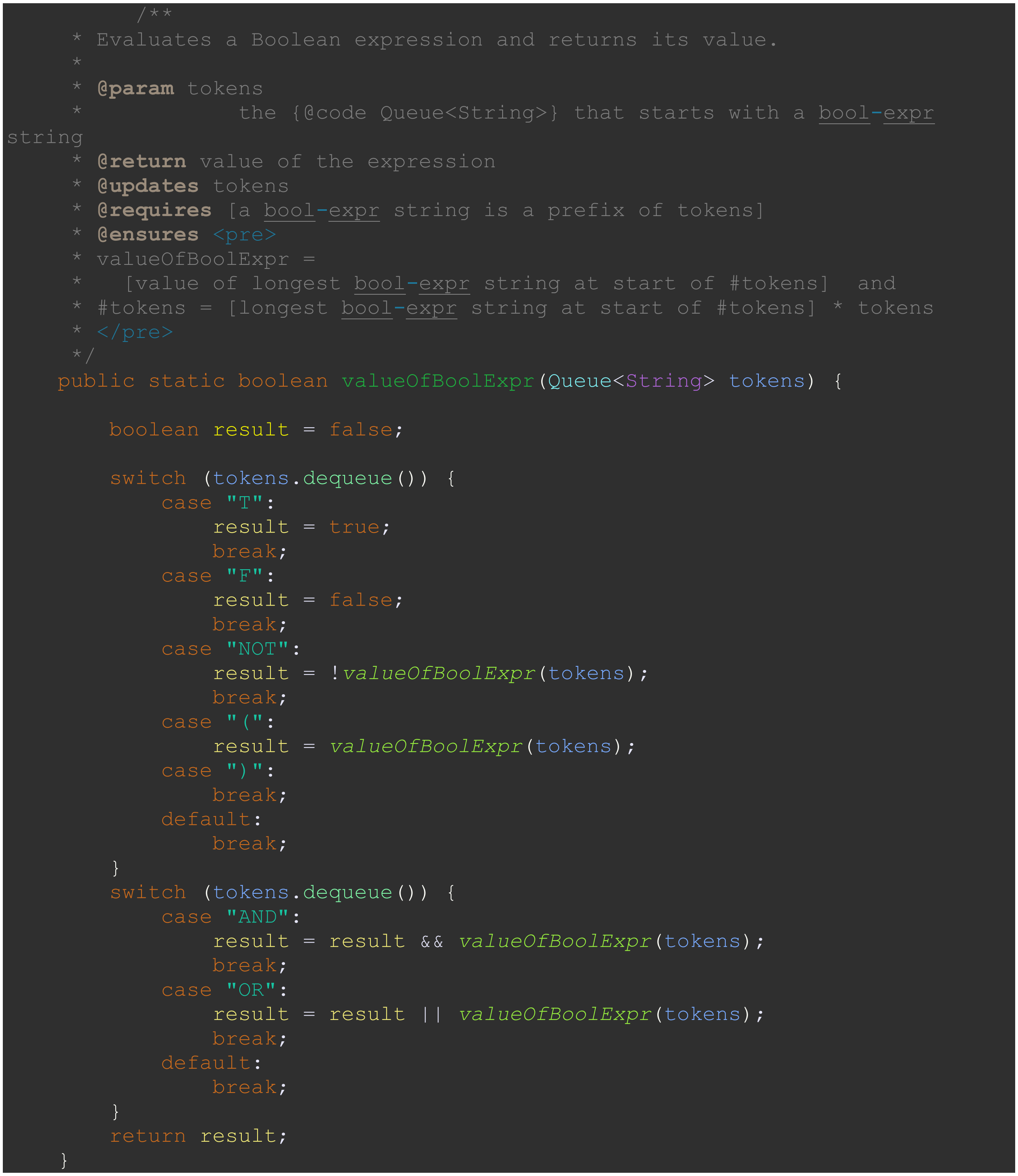
Homework 29



```
          /**
     * Evaluates a Boolean expression and returns its value.
     *
     * @param tokens
     *            the {@code Queue<String>} that starts with a bool-expr
string
     * @return value of the expression
     * @updates tokens
     * @requires [a bool-expr string is a prefix of tokens]
     * @ensures <pre>
     * valueOfBoolExpr =
     *   [value of longest bool-expr string at start of #tokens]  and
     * #tokens = [longest bool-expr string at start of #tokens] * tokens
     * </pre>
     */
    public static boolean valueOfBoolExpr(Queue<String> tokens) {

        boolean result = false;

        switch (tokens.dequeue()) {
            case "T":
                result = true;
                break;
            case "F":
                result = false;
                break;
            case "NOT":
                result = !valueOfBoolExpr(tokens);
                break;
            case "(":
                result = valueOfBoolExpr(tokens);
            case ")":
                break;
            default:
                break;
        }
        switch (tokens.dequeue()) {
            case "AND":
                result = result && valueOfBoolExpr(tokens);
                break;
            case "OR":
                result = result || valueOfBoolExpr(tokens);
                break;
            default:
                break;
        }
        return result;
    }
```