```
// A Groovy file with intentional security and syntax issues for code review
practice

class BadCode {

    def static insecureMethod() {
        def userInput = "some_user_input"
        // Security Issue 1: Command injection vulnerability
        // This is a classic example of an insecure shell command execution.
        def cmd = "ls -l " + userInput
        def proc = cmd.execute()
        println "Executing: " + cmd
        proc.waitFor()
    }

    def static vulnerableSql() {
        def id = 123
        def name = "some_name"
        // Security Issue 2: SQL injection vulnerability
        // The SQL query is constructed by concatenating strings directly,
        // making it vulnerable to injection attacks.
        def sql = "SELECT * FROM users WHERE id = " + id + " AND name = '" +
name + "'"
        println "Executing SQL: " + sql
    }

    def static insecureFileAccess() {
        def fileName = "../../../../etc/passwd"
        // Security Issue 3: Directory traversal vulnerability
        // The application accepts a filename from user input without
sanitization,
        // allowing access to sensitive files.
        File file = new File(fileName)
        if (file.exists()) {
            println "File found: " + file.getCanonicalPath()
            // In a real scenario, this would read the file's contents.
        }
    }

    // A method with multiple syntax errors
    def static syntaxErrorMethod(a, b) {
        // Syntax Issue 1: Missing semicolon
        def result = a + b
        // Syntax Issue 2: Incorrect keyword usage ( 'return' is misspelled)
        reutrn result

        // Syntax Issue 3: Unmatched parenthesis
        println("This is a syntax error"
```

```
    }

    // Method with logical and style issues
    def static confusingCode(boolean flag) {
        if (flag == true) { // Style Issue 1: Redundant comparison
            println "Flag is true"
        }
        else {
            println "Flag is false"
        }

        // Style Issue 2: Unused variable
        def unusedVariable = 100
    }

    static void main(String[] args) {
        insecureMethod()
        vulnerableSql()
        insecureFileAccess()
        syntaxErrorMethod(1, 2)
        confusingCode(true)
    }
}
```