

```
1 import java.io.*;
2 import java_cup.runtime.*;
3
4 /****
5 *
6 * Test program for CSC 330 Assignment 4. The main method constructs an
7 * EJayLexer with a FileReader. It then constructs an EJayParser, sending it
8 * the lexer. Then it calls EJayParser.parse to obtain a parse tree and symbol
9 * table for the input program. Then it constructs an EJayInterpreter,
10 * sending it the memory size for the symbol table and a stack size of 10000.
11 * Next, the test program constructs a parse tree for a parameterless
12 * invocation of the program's main function, and sends that tree to the
13 * interpreter's eval method. Finally, the test program dumps out the
14 * interpreter's memory after program execution.
15 *
16 */
17 public class EJayInterpreterTest {
18
19     /**
20     * See the class comment for documentation.
21     */
22     public static void main(String[] args) {
23         TreeNode tree;
24         SymbolTable symtab = null; // Assignment to null to such up javac
25         try {
26             EJayParser parser = new EJayParser(
27                 new EJayLexer(new FileReader(args[0])));
28
29             parser.initSymbolTable(500);
30             tree = (TreeNode) parser.parse().value;
31             symtab = parser.getSymbolTable();
32
33             EJayInterpreter interp =
34                 new EJayInterpreter(symtab.memorySize, 10000);
35             interp.eval(buildMainCall(), symtab);
36             interp.dumpMemory();
37         }
38         catch (Exception e) {
39             System.out.println("Exception " + e);
40             e.printStackTrace();
41         }
42     }
43
44     /**
45     * Construct the parse tree to invoke the main method.
46     */
47     protected static TreeNode buildMainCall() {
48         return new TreeNode2(sym.LEFT_PAREN,
49             new LeafNode(sym.VOID, "main"), null);
50     }
51
52 }
```