***Name: Kamil Roginski  
Project: CMSC 315 Programming Project 3  
Date: April 23, 2025***

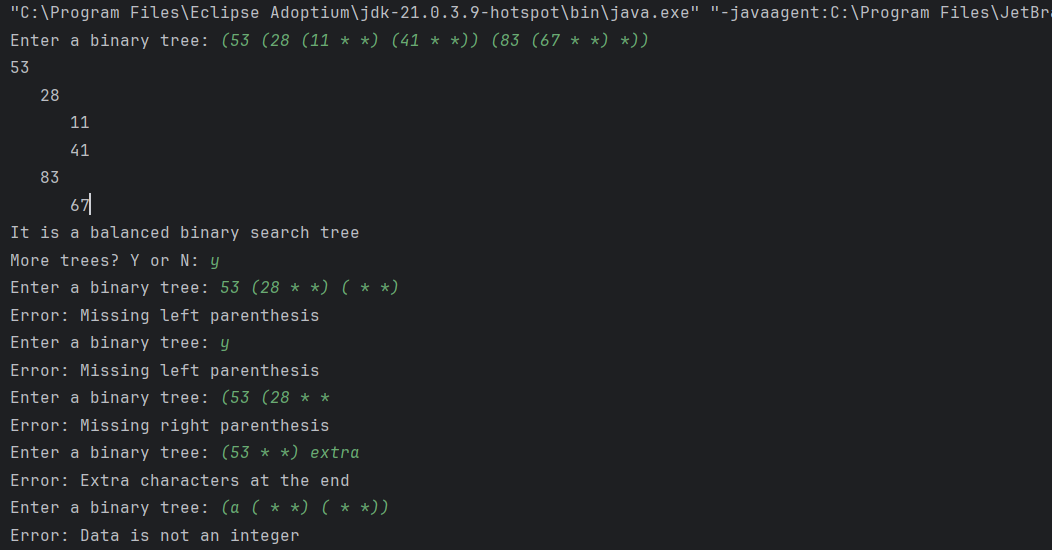
**Lessons Learned**

Working on this project taught me the importance of precise parsing and careful boundary checks. I initially forgot to call skipWhitespace() before matching the right parenthesis, which led to intermittent “Missing right parenthesis” errors on inputs like (3 (2 \* \*) (4 \* \*)) . Adding whitespace skips before each bracket check resolved that. In my first draft of checkBST(), I used n.value < min || n.value > max instead of the correct <=/>=, which allowed out-of-range values to slip through. Fixing those comparisons fixed my BST validation. I also discovered that treating an empty subtree’s height as 0 broke my balance checks, so I changed height(null) to –1 and updated the BalInfo logic accordingly.

**Test Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case** | **Input** | **Expected Outcome** | **Actual Outcome** |
| **Test Case 1 (Balanced BST)** | (53 (28 (11 \* \*) (41 \* \*)) (83 (67 \* \*) \*)) | Prints indented tree; message: “It is a balanced binary search tree” | Prints indented tree; message: “It is a balanced binary search tree” |
| **Test Case 2 (Missing Left Parenthesis)** | 53 (28 \* \*) ( \* \*) | Error: “Missing left parenthesis” | Syntax: missing leading ( |
| **Test Case 3 (Missing Right Parenthesis)** | (53 (28 \* \* | Error: “Missing right parenthesis” | Error: “Missing right parenthesis” |
| **Test Case 4 (Extra Characters at End)** | (53 \* \*) extra | Error: “Extra characters at the end” | Error: “Extra characters at the end” |
| **Test Case 5 (Data Not Integer)** | (a ( \* \*) ( \* \*)) | Error: “Data is not an integer” | Error: “Data is not an integer” |

**Test Case 1-5:**



**UML Diagram**

A screenshot of a computer program

AI-generated content may be incorrect.