

1. Error Productions

Two factors contributed to the modification of error productions within the program four parser from the program three parser. The productions made use of in this implementation of the parser fall on the continue to fall on the more coarse end of the spectrum due to the reasoning by Levine presented in the **Flex & Bison** book, though three classes of errors are provided to further clarify errors in the more complex grammar of program four.

The other classes of errors attempt to identify and direct the programmer to the nearest expression, statement, or class in which an error occurs. These three structures make up the bulk of syntactic structure within the Decaf programming language, and so being able to identify which level of the program to focus on is potentially more useful than the parsing style used in the previous assignment.

This coarse approach ensures that the correct line is identified by using the semicolon as a synchronizing token, but avoid further interpretation on the input stream to avoid pointing to a potential problem that may be incorrectly diagnosed. Thus, these errors may continue to identify larger, structural issues while pointing the programmer to individual lines with the goal of clarifying their intentions rather than guessing at their motivation for a specific line.