ATCS – Compilers and Interpreters (Peer Review Sheet)

Name Alex Hu

Date March 7, 2022 Period 3

Reviewed by Varun Fuloria Total \_\_\_\_\_\_100/100\_\_\_\_\_\_\_\_\_

**Parser Lab**

**Focus :** Use a Parse Tree like call structure to validate the input program and confirm that it is well formed as per the given grammar.

|  |  |
| --- | --- |
| Description of check to be performed. | Points/Possible Points |
| Parser Class : Comments include name, date and summary using JavaDoc Standards. | 4 / 4 |
| All method have Javadoc headers including parameters, returns, pre and post conditions and description and it all makes sense. | 8 / 8 |
| Has a package structure with a scanner and parser package. | 4 / 4 |
| Naming convention is followed package names are in lower camel case and Class names in upper camel case. | 4 / 4 |
| The current token is stored in an instance variable and is not passed to methods unnecessarily. | 2 / 2 |
| Scanner works well <=, >= , <> and := are returned as a single token. | 6 / 6 |
| A period (.) signifies end of file and if it is missing a token with value period . or END or **EOF** (EOF is preferred.) is returned when EOF is reached. The hasNext() method returns false at EOF | 6 / 6 |
| parseTerm, parseFactor, parseExpression and parseNumber are written and work well.  parseStatement handles at least 2 cases: writeln (display), assignment (and if and while/for) | 35 /35 |
| Handles Blocks of Statements (Statements enclosed in BEGIN END) | 8 / 8 |
| Code is structured well, has modularity, does not repeat blocks of code unnecessarily. | 8 / 8 |
| Testing : Works well on parserTest1, parserTest2 and parserTest3 | 15/15 |