Spectra Language

SER 502-Team 11

Fall 2024

Spectra Creators:

1. SUMIT SINGH BHADOURIA (1233009533)
2. PRERANA MADHUKAR NALE (1233568468)
3. RAGLAND PAKIYARAJ RAJASINGH (1232696610)
4. DHAARANI GANESH THANGAM (1234397998)

Copyright © 2023 Arizona Board of Regents

Overview

Copyright © 2023 Arizona Board of Regents

1.

INTRODUCTION

Copyright © 2023 Arizona Board of Regents

1. Cross-platform language designed for Windows and macOS:
Spectra is built to operate seamlessly on Windows and macOS systems, making it a versatile tool for developers across platforms.

2. Follows a complete compilation pipeline:
The development process includes all critical phases, from lexical analysis (breaking down code into tokens) to parsing (syntax validation) and runtime execution (running the code on a virtual machine).

3. Java-based implementation for portability and performance:
The language leverages Java for its runtime, ensuring compatibility across multiple operating systems and delivering efficient performance.

4. Developed using ANTLR4 for lexical analysis and parse tree generation:
ANTLR4 enables Spectra to tokenize source code into meaningful units and generate parse trees that simplify syntax analysis and semantic processing.

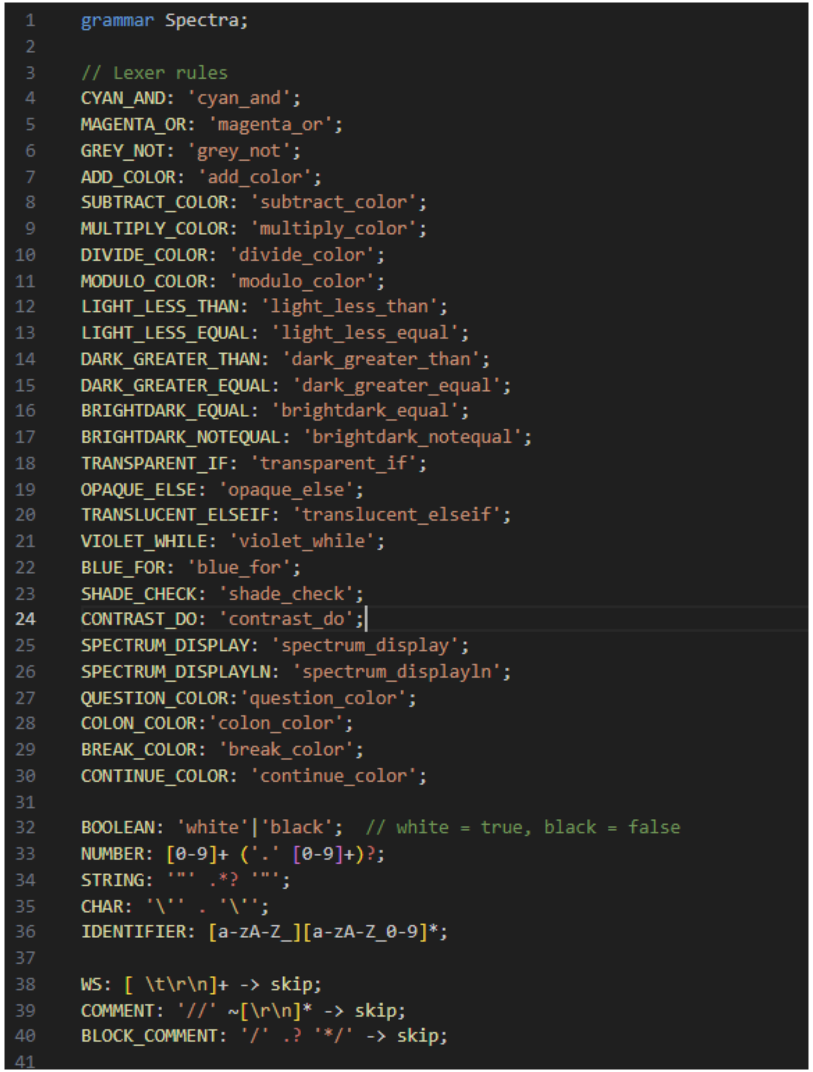
5. Inspired by imperative programming languages with robust syntax and features:
Spectra's design incorporates familiar constructs from imperative languages, such as loops, conditionals, and operators, providing an intuitive and robust programming experience.

Copyright © 2023 Arizona Board of Regents

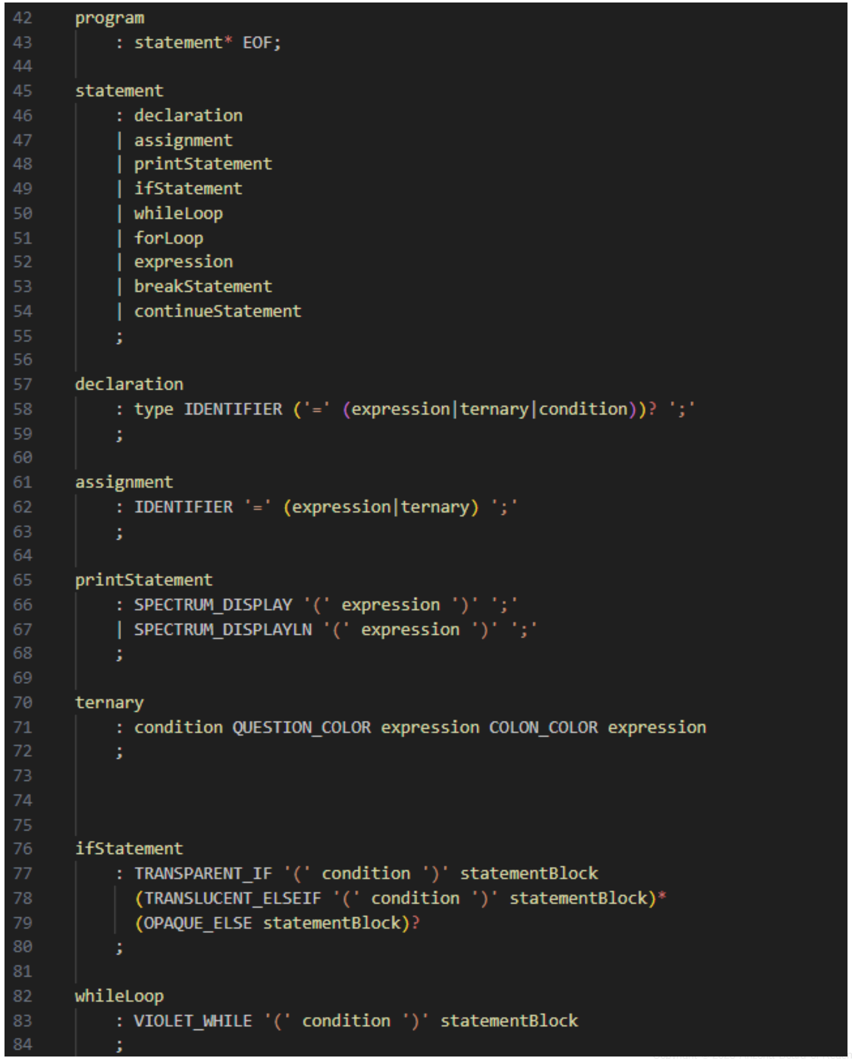
2.

GRAMMAR

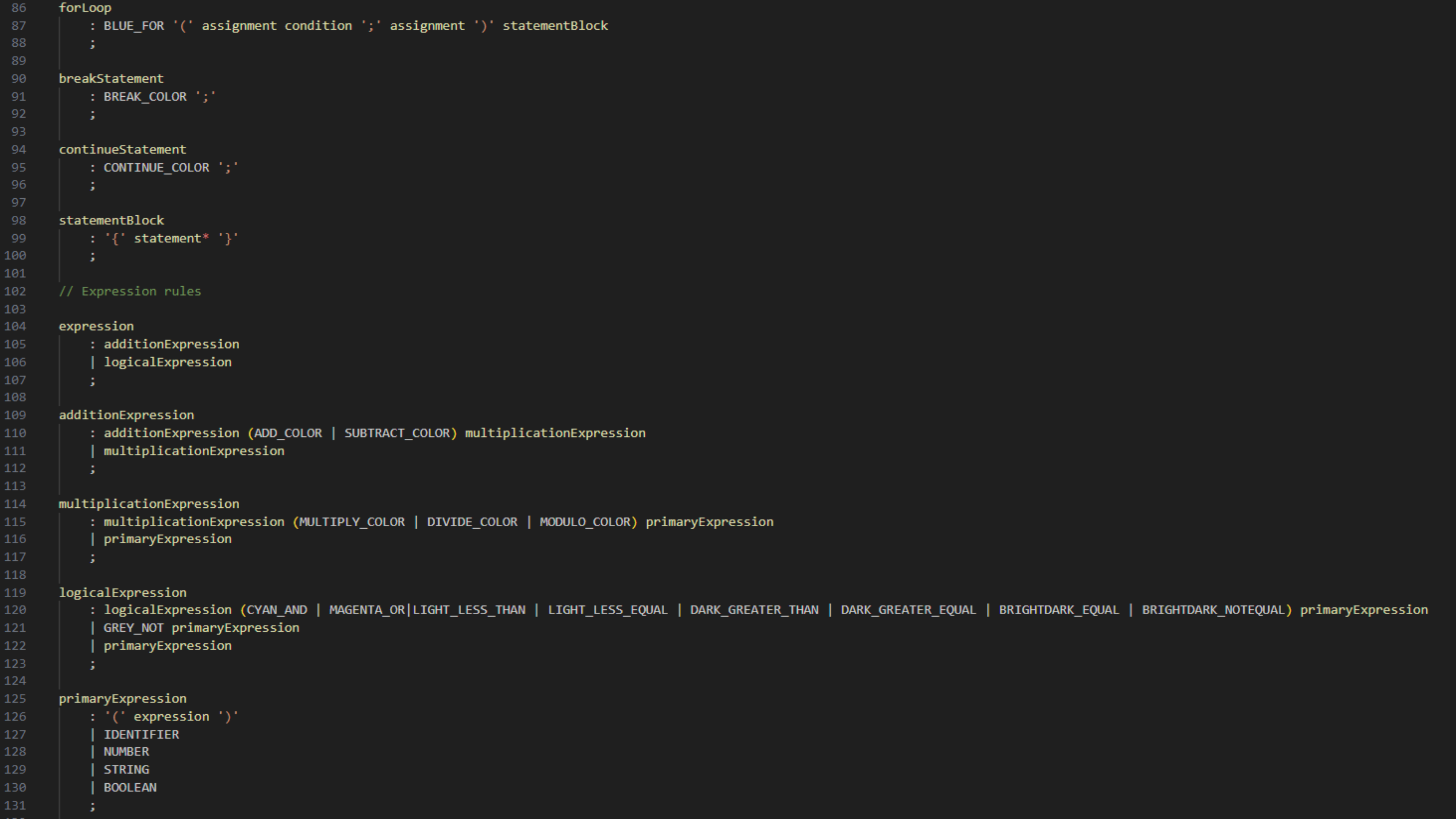
Copyright © 2023 Arizona Board of Regents



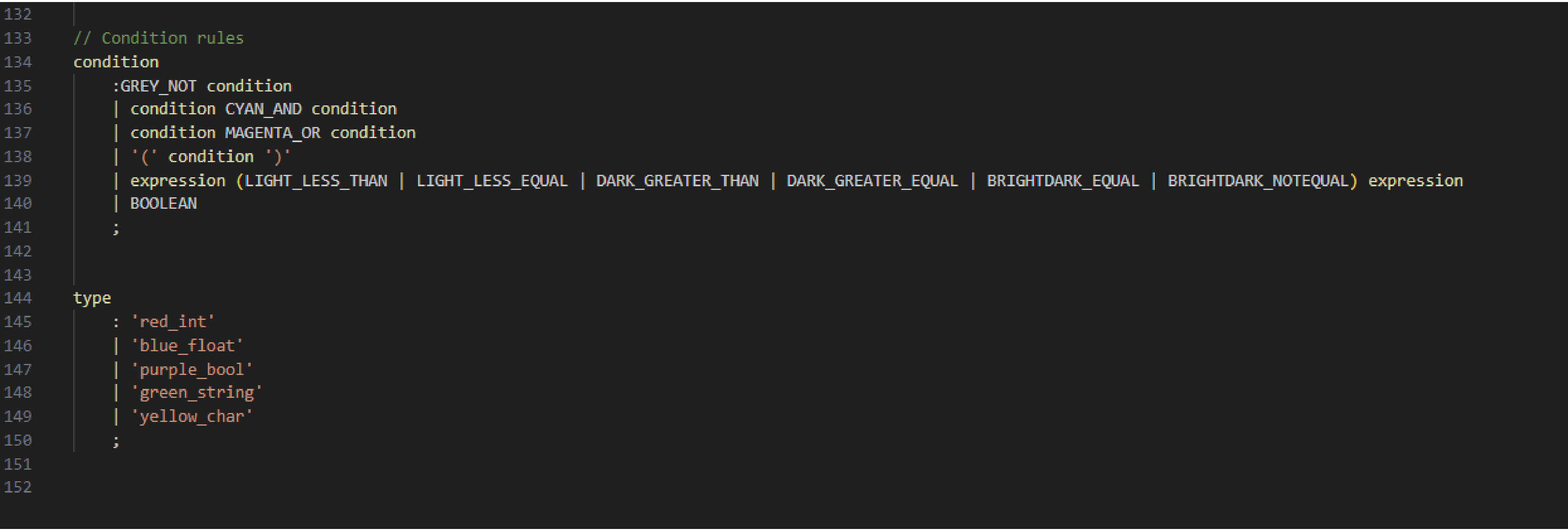
```
grammar Spectra;

// Lexer rules
CYAN_AND: 'cyan_and';
MAGENTA_OR: 'magenta_or';
GREY_NOT: 'grey_not';
ADD_COLOR: 'add_color';
SUBTRACT_COLOR: 'subtract_color';
MULTIPLY_COLOR: 'multiply_color';
DIVIDE_COLOR: 'divide_color';
MODULO_COLOR: 'modulo_color';
LIGHT_LESS_THAN: 'light_less_than';
LIGHT_LESS_EQUAL: 'light_less_equal';
DARK_GREATER_THAN: 'dark_greater_than';
DARK_GREATER_EQUAL: 'dark_greater_equal';
BRIGHTDARK_EQUAL: 'brightdark_equal';
BRIGHTDARK_NOTEQUAL: 'brightdark_notequal';
TRANSPARENT_IF: 'transparent_if';
OPAQUE_ELSE: 'opaque_else';
TRANSLUCENT_ELSEIF: 'translucent_elseif';
VIOLET_WHILE: 'violet_while';
BLUE_FOR: 'blue_for';
SHADE_CHECK: 'shade_check';
CONTRAST_DO: 'contrast_do';
SPECTRUM_DISPLAY: 'spectrum_display';
SPECTRUM_DISPLAYLN: 'spectrum_displayln';
QUESTION_COLOR:'question_color';
COLON_COLOR:'colon_color';
BREAK_COLOR: 'break_color';
CONTINUE_COLOR: 'continue_color';

BOOLEAN: 'white'|'black';  // white = true, black = false
NUMBER: [0-9]+ ('.' [0-9]+)?;
STRING: '"' .*? '"';
CHAR: '\'' . '\'';
IDENTIFIER: [a-zA-Z_][a-zA-Z_0-9]*;

WS: [ \t\r\n]+ -> skip;
COMMENT: '//' ~[\r\n]* -> skip;
BLOCK_COMMENT: '/' .? '*/' -> skip;

```



```
program
    : statement* EOF;

statement
    : declaration
    | assignment
    | printStatement
    | ifStatement
    | whileLoop
    | forLoop
    | expression
    | breakStatement
    | continueStatement
    ;

declaration
    : type IDENTIFIER ('=' (expression|ternary|condition))? ';'
    ;

assignment
    : IDENTIFIER '=' (expression|ternary) ';'
    ;

printStatement
    : SPECTRUM_DISPLAY '(' expression ')' ';'
    | SPECTRUM_DISPLAYLN '(' expression ')' ';'
    ;

ternary
    : condition QUESTION_COLOR expression COLON_COLOR expression
    ;



ifStatement
    : TRANSPARENT_IF '(' condition ')' statementBlock
      (TRANSLUCENT_ELSEIF '(' condition ')' statementBlock)*
      (OPAQUE_ELSE statementBlock)?
    ;

whileLoop
    : VIOLET_WHILE '(' condition ')' statementBlock
    ;
```



```
forLoop
    : BLUE_FOR '(' assignment condition ';' assignment ')' statementBlock
    ;

breakStatement
    : BREAK_COLOR ';'
    ;

continueStatement
    : CONTINUE_COLOR ';'
    ;

statementBlock
    : '{' statement* '}'
    ;

// Expression rules

expression
    : additionExpression
    | logicalExpression
    ;

additionExpression
    : additionExpression (ADD_COLOR | SUBTRACT_COLOR) multiplicationExpression
    | multiplicationExpression
    ;

multiplicationExpression
    : multiplicationExpression (MULTIPLY_COLOR | DIVIDE_COLOR | MODULO_COLOR) primaryExpression
    | primaryExpression
    ;

logicalExpression
    : logicalExpression (CYAN_AND | MAGENTA_OR|LIGHT_LESS_THAN | LIGHT_LESS_EQUAL | DARK_GREATER_THAN | DARK_GREATER_EQUAL | BRIGHTDARK_EQUAL | BRIGHTDARK_NOTEQUAL) primaryExpression
    | GREY_NOT primaryExpression
    | primaryExpression
    ;

primaryExpression
    : '(' expression ')'
    | IDENTIFIER
    | NUMBER
    | STRING
    | BOOLEAN
    ;
```



```
// Condition rules
condition
    :GREY_NOT condition
    | condition CYAN_AND condition
    | condition MAGENTA_OR condition
    | '(' condition ')'
    | expression (LIGHT_LESS_THAN | LIGHT_LESS_EQUAL | DARK_GREATER_THAN | DARK_GREATER_EQUAL | BRIGHTDARK_EQUAL | BRIGHTDARK_NOTEQUAL) expression
    | BOOLEAN
    ;


type
    : 'red_int'
    | 'blue_float'
    | 'purple_bool'
    | 'green_string'
    | 'yellow_char'
    ;


```

Copyright © 2023 Arizona Board of Regents

Features

1. Control Structures:

- **Conditional Statements:**

➢ transparent_if for if statement

➢ opaque_else for else statement

➢ translucent_elseif for if-else logic.

- **Loops:**

➢ violet_while for while loops.

➢ blue_for for loops.

2. Print Statements:

➢ spectrum_display for output

➢ spectrum_displayln for output in the new line

3. Arithmetic Operations:

➢ Addition: add_color

➢ Subtraction: subtract_color

➢ Multiplication: multiply_color

➢ Division: divide_color

➢ Modulus: modulo_color

Copyright © 2023 Arizona Board of Regents

4. Logical Operations:

- Logical AND: cyan_and
- Logical OR: magenta_or
- Logical NOT: grey_not

5. Comparison Operations:

- Less than: light_less_than
- Less than or equal: light_less_equal
- Greater than: dark_greater_than
- Greater than or equal: dark_greater_equal
- Equality: brightdark_equal
- Inequality: brightdark_notequal

6.Break and Continue: (for controlling loop execution)

- Break: break_color
- Continue: continue_color

Copyright © 2023 Arizona Board of Regents

7. Data Types:

- red_int: Represents integers.
- blue_float: Represents floating-point numbers.
- purple_bool: Represents boolean values (white for true and black for false).
- green_string: Represents strings.
- yellow_char: Represents single characters.

8. Ternary Operator:

- question_color :Represents ?
- colon_color : Represents :

Copyright © 2023 Arizona Board of Regents

Variable Naming Convention

- A variable name must start with a letter (a-z, A-Z) or an underscore (_).
- It can be followed by letters, digits (0-9), or underscores.
- Example: Valid variable names include var1, _temp, and color_variable.

Copyright © 2023 Arizona Board of Regents

Reserved Keywords

- cyan_and
- magenta_or
- grey_not
- add_color
- subtract_color
- multiply_color
- divide_color
- modulo_color
- light_less_than
- light_less_equal
- dark_greater_than
- dark_greater_equal
- brightdark_equal
- brightdark_notequal
- transparent_if
- opaque_else
- translucent_elseif
- violet_while
- blue_for
- shade_check
- contrast_do
- spectrum_display
- spectrum_displayln
- question_color
- colon_color
- break_color
- continue_color
- white (true boolean value)
- black (false boolean value)
- red_int
- blue_float
- purple_bool
- green_string
- yellow_char

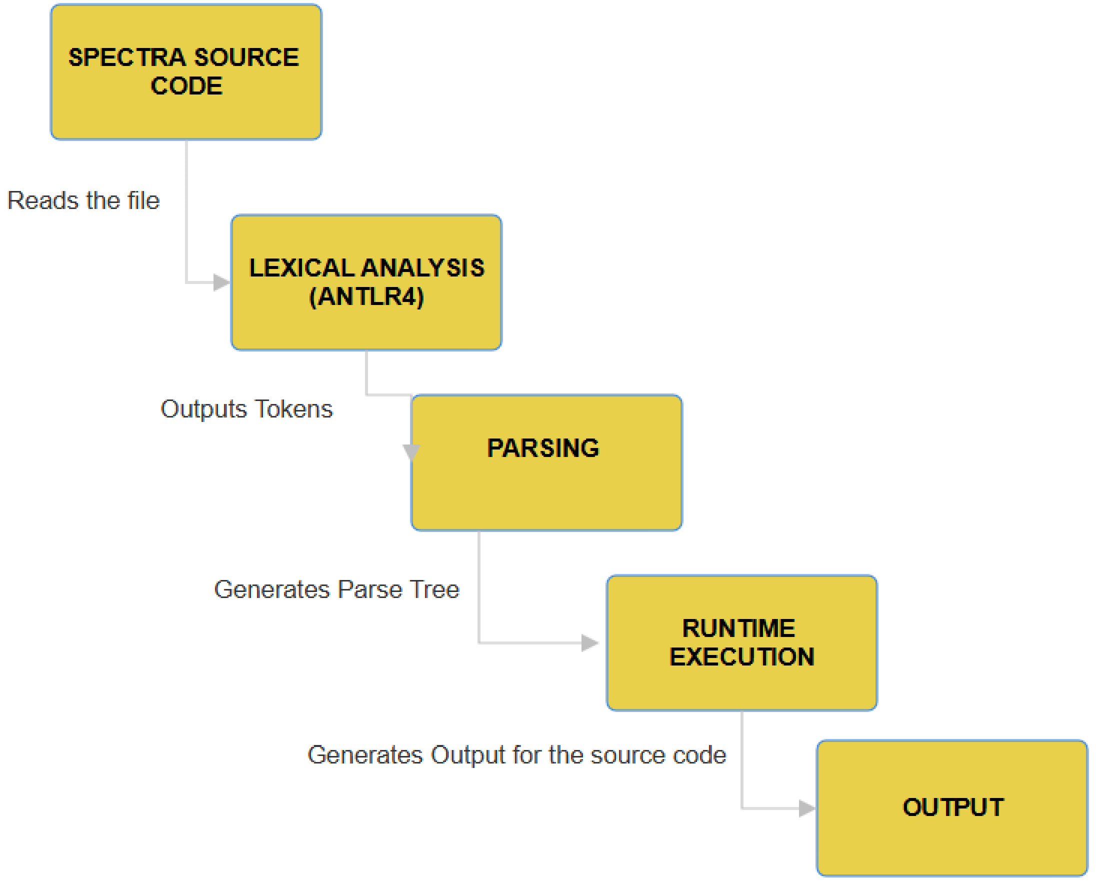
Copyright © 2023 Arizona Board of Regents

3.

COMPILER

Copyright © 2023 Arizona Board of Regents

DESIGN FLOW



Copyright © 2023 Arizona Board of Regents

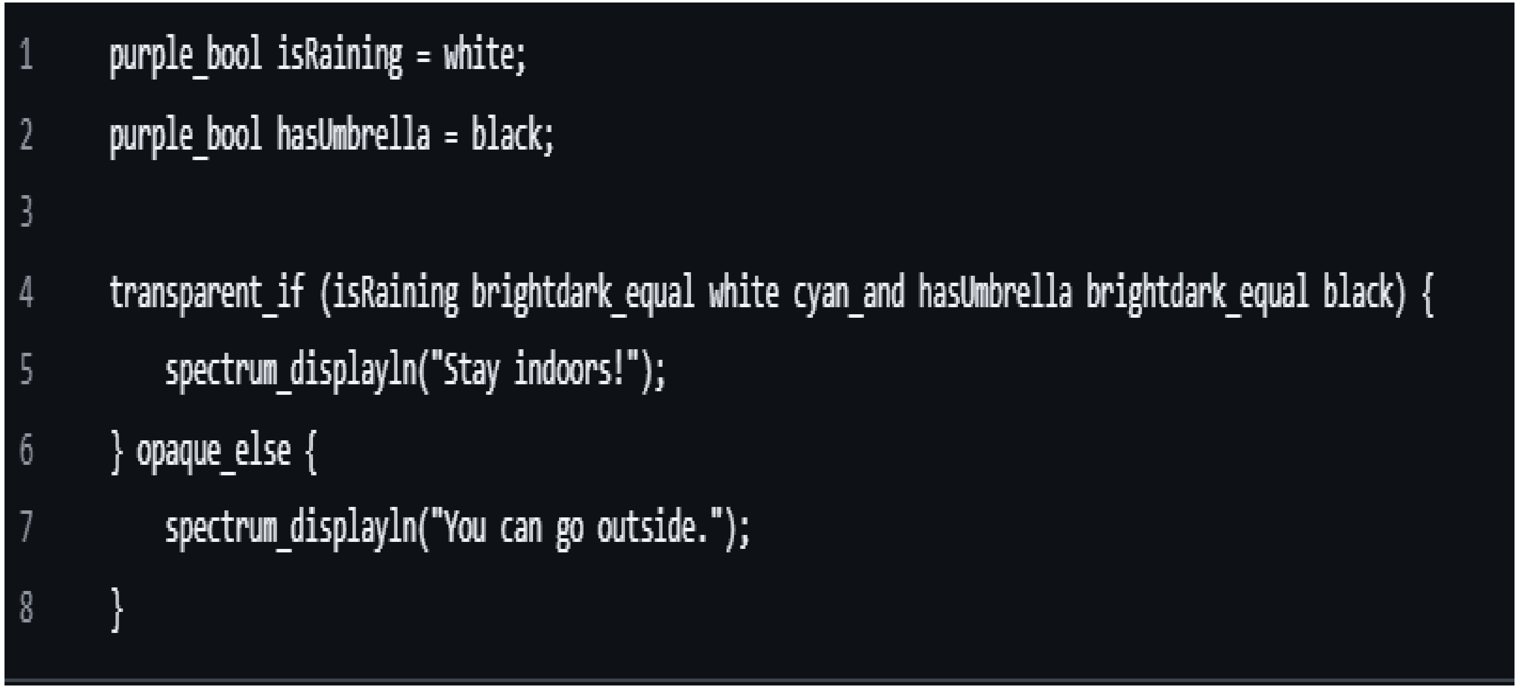
4.

SAMPLE CODE

Copyright © 2023 Arizona Board of Regents

Spectra File should be saved with the extension: “.spc”
Example: File.spc

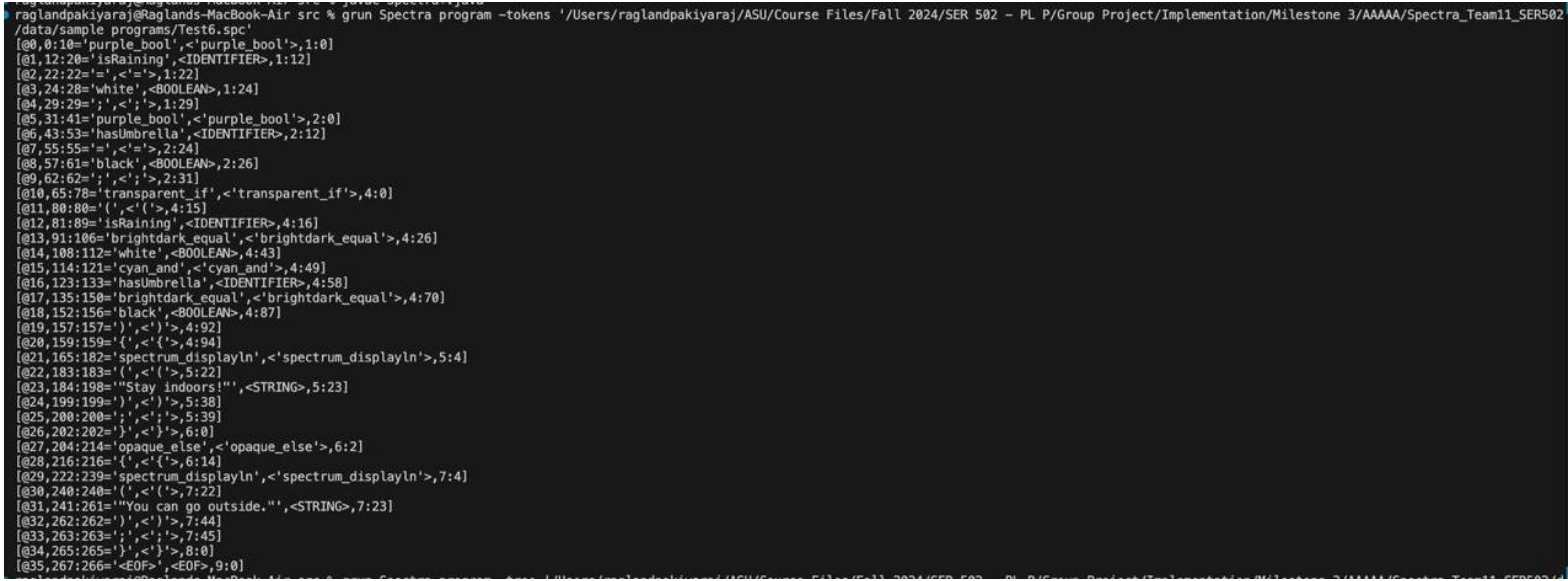
SAMPLE CODE:



```
purple_bool isRaining = white;
purple_bool hasUmbrella = black;

transparent_if (isRaining brightdark_equal white cyan_and hasUmbrella brightdark_equal black) {
    spectrum_displayln("Stay indoors!");
} opaque_else {
    spectrum_displayln("You can go outside.");
}
```

Copyright © 2023 Arizona Board of Regents

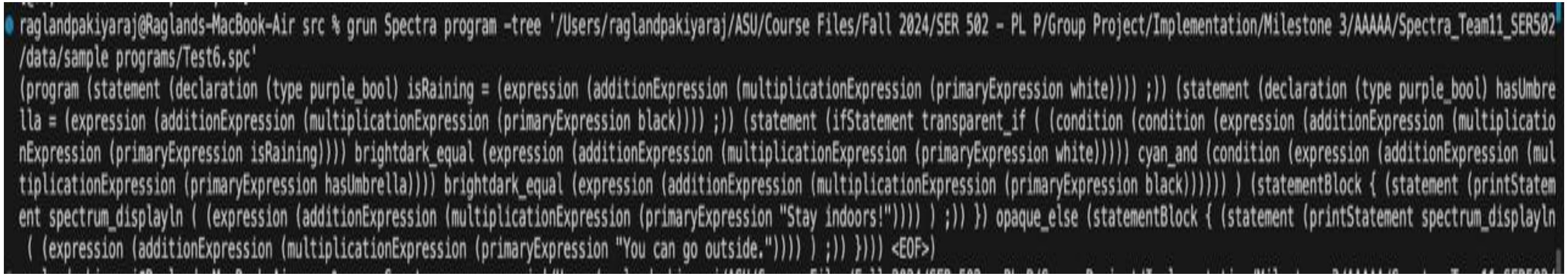
TOKENS



```
raglandpakiyaraj@Raglands-MacBook-Air src % grun Spectra program -tokens '/Users/raglandpakiyaraj/ASU/Course Files/Fall 2024/SER 502 - PL P/Group Project/Implementation/Milestone 3/AAAAA/Spectra_Team11_SER502
/data/sample programs/Test6.spc'
[@0,0:10='purple_bool',<'purple_bool'>,1:0]
[@1,12:20='isRaining',<IDENTIFIER>,1:12]
[@2,22:22='=',<'='>,1:22]
[@3,24:28='white',<BOOLEAN>,1:24]
[@4,29:29=';',<';'>,1:29]
[@5,31:41='purple_bool',<'purple_bool'>,2:0]
[@6,43:53='hasUmbrella',<IDENTIFIER>,2:12]
[@7,55:55='=',<'='>,2:24]
[@8,57:61='black',<BOOLEAN>,2:26]
[@9,62:62=';',<';'>,2:31]
[@10,65:78='transparent_if',<'transparent_if'>,4:0]
[@11,80:80='(',<'('>,4:15]
[@12,81:89='isRaining',<IDENTIFIER>,4:16]
[@13,91:106='brightdark_equal',<'brightdark_equal'>,4:26]
[@14,108:112='white',<BOOLEAN>,4:43]
[@15,114:121='cyan_and',<'cyan_and'>,4:49]
[@16,123:133='hasUmbrella',<IDENTIFIER>,4:58]
[@17,135:150='brightdark_equal',<'brightdark_equal'>,4:70]
[@18,152:156='black',<BOOLEAN>,4:87]
[@19,157:157=')',<')'>,4:92]
[@20,159:159='{',<'{'>,4:94]
[@21,165:182='spectrum_displayln',<'spectrum_displayln'>,5:4]
[@22,183:183='(',<'('>,5:22]
[@23,184:198='"Stay indoors!"',<STRING>,5:23]
[@24,199:199=')',<')'>,5:38]
[@25,200:200=';',<';'>,5:39]
[@26,202:202='}',<'}'>,6:0]
[@27,204:214='opaque_else',<'opaque_else'>,6:2]
[@28,216:216='{',<'{'>,6:14]
[@29,222:239='spectrum_displayln',<'spectrum_displayln'>,7:4]
[@30,240:240='(',<'('>,7:22]
[@31,241:261='"You can go outside."',<STRING>,7:23]
[@32,262:262=')',<')'>,7:44]
[@33,263:263=';',<';'>,7:45]
[@34,265:265='}',<'}'>,8:0]
[@35,267:266='<EOF>',<EOF>,9:0]
```

Copyright © 2023 Arizona Board of Regents

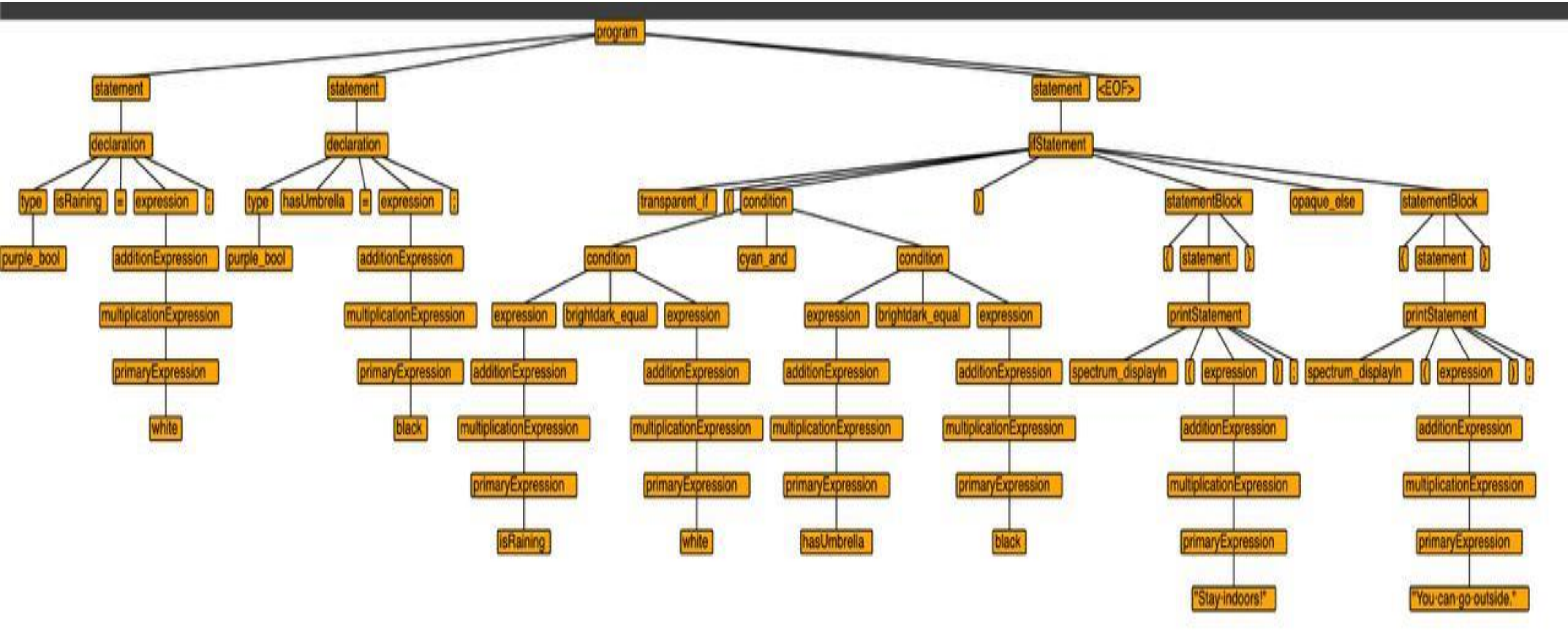
PARSE TREE



```
raglandpakiyaraj@Raglands-MacBook-Air src % grun Spectra program -tree '/Users/raglandpakiyaraj/ASU/Course Files/Fall 2024/SER 502 - PL P/Group Project/Implementation/Milestone 3/AAAAA/Spectra_Team11_SER502
/data/sample programs/Test6.spc'
(program (statement (declaration (type purple_bool) isRaining = (expression (additionExpression (multiplicationExpression (primaryExpression white)))) ;)) (statement (declaration (type purple_bool) hasUmbre
lla = (expression (additionExpression (multiplicationExpression (primaryExpression black)))) ;)) (statement (ifStatement transparent_if ( (condition (condition (expression (additionExpression (multiplicatio
nExpression (primaryExpression isRaining)))) brightdark_equal (expression (additionExpression (multiplicationExpression (primaryExpression white))))) cyan_and (condition (expression (additionExpression (mul
tiplicationExpression (primaryExpression hasUmbrella)))) brightdark_equal (expression (additionExpression (multiplicationExpression (primaryExpression black)))))) ) (statementBlock { (statement (printStatem
ent spectrum_displayln ( (expression (additionExpression (multiplicationExpression (primaryExpression "Stay indoors!")))) ) ;)) }) opaque_else (statementBlock { (statement (printStatement spectrum_displayln
 ( (expression (additionExpression (multiplicationExpression (primaryExpression "You can go outside.")))) ) ;)) }))) <EOF>)
```

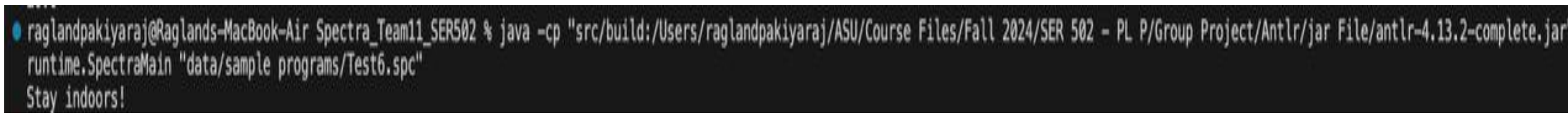
Copyright © 2023 Arizona Board of Regents

GUI OF PARSE TREE



Copyright © 2023 Arizona Board of Regents

OUTPUT



```
raglandpakiyaraj@Raglands-MacBook-Air Spectra_Team11_SER502 % java -cp "src/build:/Users/raglandpakiyaraj/ASU/Course Files/Fall 2024/SER 502 - PL P/Group Project/Antlr/jar File/antlr-4.13.2-complete.jar
runtime.SpectraMain "data/sample programs/Test6.spc"
Stay indoors!
```

Copyright © 2023 Arizona Board of Regents

5.

FUTURE SCOPE

Copyright © 2023 Arizona Board of Regents

FUTURE SCOPE:

- **Data Structures:**
  Include Data Structures like array, list, and dictionary to further enhance the capability of the language.

- **User-Defined Functions:**
  Allow the definition of user-defined functions for modular programming and reuse of code.

- **Error Handling:**
  Include structured error handling mechanisms such as try-catch for better debugging and runtime safety.

- **Concurrency Support:**
  Add support for multi-threading and asynchronous operations to efficiently handle concurrent tasks in a program.

Copyright © 2023 Arizona Board of Regents

THANK YOU

Arizona State University

Copyright © 2023 Arizona Board of Regents