CPSC 410 A1

Student ID: 55788673

Exercise 1

The approach started parsing by building top node of the AST tree from the start nonterminal. The approach also converts the whole program into a single string and read the token from left to right. There is no left recursion or backtracking involved in the approach.

Exercise 2

```
public void parse(){
        // Parse Title
        if (tokenizer.checkToken("Titile")) {
            title = new TITLE();
            title.parse();
        } else {
            title = new TITLE();
            title.title = "";
        }

        // Parse Tables
        while(tokenizer.moreTokens()){
            TABLE t = new TABLE();
            t.parse();
            tables.add(t);
        }
    }
```

Exercise 3

Yes, the modified grammar is ambiguous. The reason for that is we have no way to identify if the next string is table or title. For example, if title starts with "[", then we are unable to distinguish them.

Exercise 4

```
public class Shape {
    private String base;
    private Shape inside = new Shape();
    public void parse() {
        if (tokenizer.checkToken("Triangle") == true) {
            tokenizer.getNext();
            base = "Triangle";
        }
        if (tokenizer.checkToken("Circle") == true) {
            tokenizer.getNext();
```

```
            base = "Circle";
        }
        tokenizer.getAndCheckNext("With");
        while(tokenizer.moreTokens() && !tokenizer.checkToken("Inside")) {
            inside.parse();
        }
        tokenizer.getAndCheckNext("Inside");
    }
}
```

Exercise 5

Grammar 1

SHAPE ::= (COLOR) (“Circle” | “Triangle” | “Rectangle”)

COLOR ::= "Blue" | "Red" | "Green"

Grammar 2

SHAPE ::= (“Blue” (“ Triangle” | “Circle”)) | ((“Green” | “ Red”) “ Circle)

COLOR ::= "Blue" | "Red" | "Green"

Grammar 3

SHAPE ::= (“Triangle” “Circle”) | (“Circle” “Circle”?)

COLOR ::= "Blue" | "Red" | "Green"

Explanation:

Our factorization is to make the left most token be clear about what path should we follow. If we start with “Triangle”, then there must be a “ Circle” after it. If we start with “Circle”, we could either have a “Circle” after it or nothing.

Grammar 4

SHAPE ::= (“Triangle” SHAPE+) | (“Circle” SHAPE*)

Explanation: The expression means we want to end with “Circle” but there are random number of “Circle” and “Triangle” before this ending. If we start with “Circle”, we could end at this point, so we use “*”, but if we start with “Triangle”, the we should repeat SHAPE to satisfy the expression ends with “Circle.