Assignment 1
Lex Design

This document introduces a new programming language JAVA--. Your task is to develop a Lexical Analyzer for this language.

The Lex shall take a text file containing JAVA-- source code as input, and shall tokenize it into token-lexeme pairs.

Following is a sample program written in JAVA--.

```
int numPrint (int num, int length)
{
        int i, j, first, temp;
        char a;
        a<-‘x’;
        jOut( "enter number");
        jIn (i);
        jOut(i);
        i <- length;
        while (i > 0)
        {
                first<- 0;              /*this line contains a comment*/
                j <-1;
                while (j < i)
                {
                        jOut( j);
                        j <- j + 1;
                }
                /* this is a comment */
                i<- i - 1;
                /*This is a
                Multiline
                Comment*/
        }
        jOut (“temp is “);
        jOut( temp);
        return i;
}
```

The language contains the following elements:
data types: int char
Keywords: if else while return jIn jOut
arithmetic operators: + - * /
relational operators: < <= > >= == !=
comments: /* enclose comment in */
identifier: a letter followed by any number of letters or digits
numeric constants: only integers
literal constants: a letter enclosed in single quotes
strings: no need to store as variables, only used in print statements

parenthesis, braces, square brackets
assignment operator <-
semi colon
colon
comma

Requirements:

Your task in this assignment is to document the language JAVA-- and design a lexical analyzer for it. There will be no implementation in this assignment. All illustrations must be digital. No photographs of hand-drawn transition diagrams will be accepted. You are required to provide:

- Complete description of language tokens (and lexemes where applicable)
- Regular definitions and transition diagrams for any complex tokens (those with more than one lexeme).

This will be the first part of the documentation of your language and compiler. Name this file documentation.pdf. No other file formats will be accepted.

Submission Instructions:
Submit files unzipped
Do not submit executables.
All deliverables will be marked without an evaluation, so make all programming deliverables compile **ON Windows**.
There will be zero tolerance for plagiarism. Your assignments will be checked far more thoroughly than you are anticipating. Once detected, no appeals for removal of plagiarism will be entertained.