

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
Compiler Design Lab – CS6612

Programming Assignment-3 - Implementation of Symbol Table Construction using Lex

Due Date: 20.02.17 & 22.02.17

Develop a Lexical analyzer to recognize the patterns namely, identifiers, constants, and keywords using the following regular expressions.

Regular Expression for Identifier letter → [a-zA-Z] digit → [0-9] id → letter(letter digit)*	Regular Expression for Constants digit → [0-9] digits → digit digits optFrac → .digits optExp → E(+ - ε) digits numberconst → digits optFrac optExp charconst → '(letter)' stringconst → "(letter)*" constant → numberconst charconst stringconst
Regular Expression for keywords int → int float → float char → char double → double keywords → int float char double	

Convert the regular expressions into cumulative transition diagram. Each state represents a condition that could occur during the process of scanning the input looking for a lexeme that matches one of the several patterns. Convert each state into a piece of code. Test the code using the following test case

Input

```
int a=9, b1, number=10;
```

```
float f1=4.5, f2=6E2;
```

```
float f3=4E+9;
```

```
char c='a';
```

Output

Symbol Table

Name	Type	Value
a	int	9
b1	int	0
number	int	10
f1	float	4.5
f2	float	6E2
f3	float	4E+9
C	char	'a'