Name : Muhammad Numan Alvi

Roll No : L1F22BSCS0440

-> Regex definitions (in table form).

-> Explanation of chosen 15 keywords + operators + punctuations.

Token Type	Regex	Description
Single-line Comment	!![^'\n']*	Matches comments starting with !! till the end of the line
Multi-line Comment	"!”([^]|*[^!])*"*!"	Multi Line Comment deal easily using this
Unterminated Multi-line Comment	"!”([^]|*[^!])*$	If we are unable to end it so error occur
Valid Identifier	[@][a-zA-Z_][a-zA-Z0-9_]*	Identifier starting with @, followed by letter or underscore, then letters/digits/underscore
Invalid Identifier	[^@][a-zA-Z_][a-zA-Z0-9_]*	first character invalid (not letter or underscore)
Integer	[0-9]+	Matches integer numbers
Float	[0-9]*\.[0-9]+	Matches decimal numbers (fractional part mandatory)
String	\"([^"\n]*)"	Matches double-quoted strings
Unterminated String	\"[^"\n]*\n	Matches strings that are not closed properly
Character	\'[a-zA-Z0-9!@#$%^&*()_+=-;'.]\'	Matches valid single-character literals
Invalid Character	\'.{2,}\'	Matches character literals with more than one character
Assignment Operator	==	Used for assignment
Addition Operator	++	Increment / addition operator

Subtraction Operator	--	Decrement / subtraction operator
Multiply Operator	**	Multiplication
Divide Operator	//	Division
Modulus Operator	%%	Modulus operator
Not Equal Operator	!=	<>
And Operator	&&	Logical AND
Or Operator	||	Logical OR
Not Operator	!	Logical NOT
Less Than / Equal Operator	<=	Less than or equal
Greater Than / Equal Operator	>=	Greater than or equal
Less Than Operator	<	Less than
Greater Than Operator	>	Greater than
Equal Operator	=	Equality operator
Custom Output Operator	\(\(:"	Output operator ((:
Custom input Operator	\)\):	Input operator)):
Punctuation	[{}(),"']	Curly braces, parentheses, comma, quotes, tilde
Whitespace	[\t\r]+	Spaces, tabs, carriage returns (ignored)
Newline	\n	Increment line counter
Error / Unknown Token	.	Any unmatched character

Keyword	**Equivalent to C++**	**Example**
load	#include	Lood <iostream>
workspace	namespace	Using workspace std;

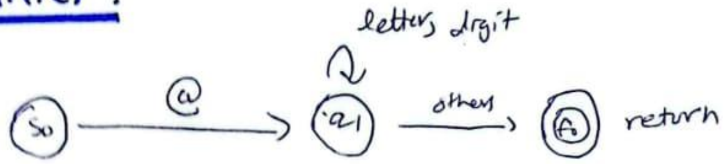
Start	int main ()	start { ... }
def	function	def draw () { }
export	cout	euport << "Design t";
import	cin	import >> color;
Unit	int	Unit width = 1000;
duplicate	double	duplicate opacity = 0.75;
text	string	text Title = "Hi";
color	string/hex	color bg = "#fff";
toggle	bool	toggle visible = true;
select	if	Select (bg == "#fff") { ... }
otherwise	else	otherwise { ... }
repeat	for	repeat (i = 0 ; i < 3 ; i++)
Loop	while	Loop (n < 5) { ... }
Save	return	Save 0;
blend	+/combine	blend color 1 + color 2;

I Choose these Keywords Because i want to make it easy for designers to learn language.

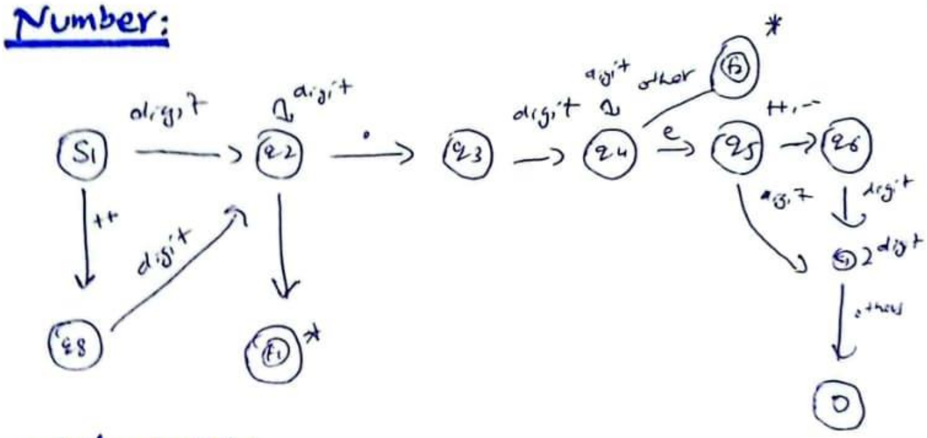
Transition diagrams for Identifiers and Numbers.

FA

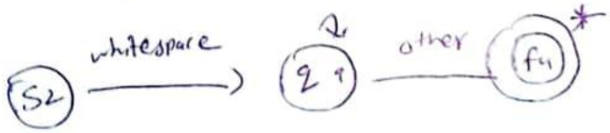
identifier:



Number:



whitespaces:



whitespace
other
S2
q9
f4
*