Lexemes and Tokens for C♭

Please refer to Lecture 4 slides for the extended regular expression syntax. The characters in orange monospace are characters that appear verbatim in the concrete syntax. The tokens are in <blue sans-serif>, surrounded with angle brackets. Any extra data stored in the token appears after the token type. For example, "<ID,string>" is a token with type "ID" and a string payload.

Lexemes	Token
[a-zA-Z][0-9a-zA-Z]*	<ID,string>
-$^?$[0-9]$^+$	<NUM,integer>
int	<TYPE,"int">
if	<IF>
else	<ELSE>
while	<WHILE>
for	<FOR>
from	<FROM>
to	<TO>
def	<DEF>
return	<RETURN>
output	<OUTPUT>
(	<LPAREN>
)	<RPAREN>
[	<LBRACKET>
]	<RBRACKET>
{	<LBRACE>
}	<RBRACE>
;	<SEMICOLON>
:=	<ASSIGN>
:	<HASTYPE>
,	<COMMA>
!	<LNEG>
+, *, -	<AOP,operator> where operator is one of +,*,-
<, <=, =	<ROP,operator> where operator is one of <,<=,=
&&, ||	<LBINOP,operator> where operator is one of and, or
[\\n\t]$^+$	whitespace, no token
//. ∗ \n	comment, no token