

Programming Project 1: Lexical Analysis

The subset of GLSL for Project.

1. All tokens highlighted in GLSL spec section 3.7 will be used in this assignment.
2. Integer, bool, and float constants are described in Section 4.1.2 – 4.1.4. We will NOT test octal-constant and hexadecimal integers. For floats, we will not test digitsequence. or .digitsequence. For example, we won't test 1. or .2 . To represent a negative number, we need to do a unary operator ("-") with that number. Boolean "true" and "false" have bool values (don't display them as 1 or 0).
3. Main operators are listed in the table in section 3.4. "++" and "--" token types are displayed as "T_Inc" and "T_Dec". All operators will be reported with proper token types defined in ***scanner.h***.
4. The rest of tokens are highlighted in Section 10.
5. Comments are described in Section 3.5.
6. There are no "char" and "string" data types in GLSL. There are no pointers in GLSL. So "this" is a reserved word, not a token. There are no class and "new/delete" operators.
7. Reserved words should not be used in GLSL program. But we won't test them (i.e. reporting an error) in this assignment.
8. GLSL identifiers can be up to 1023 letters. See the example in [sampler](#) folder how they should be handled.