

Question1:

$\wedge \{letter\}(\{letter\}|\{digit\})^*((\{letter\}|\{digit\})^*\{hyphen\}(\{letter\}|\{digit\})+)^*\#\{digit\}^+$

Question2:

1. {SLASH}{STAR}[  $\wedge$  (\* /)]\*{STAR}{SLASH} is correct pattern

2. {SLASH}{STAR}(.)\*{STAR}{SLASH} is incorrect pattern because the pattern allows the comment body to include \*/.

/\*\*/\*/ is matched by the pattern but is a “bad” comment.

3. {SLASH}{STAR}([  $\wedge$  \*] \*{STAR}+[  $\wedge$  \*/]) \*{STAR}+{SLASH} is incorrect pattern because the pattern allows the comment body to include \*/.

/\* // \*/\*\*/ is matched by the pattern but is a “bad” comment.

4. {SLASH}{STAR}([  $\wedge$  \*] | [  $\wedge$  /])+{STAR}{SLASH} is incorrect pattern because the pattern allows the comment body to include \*/.

/\* /\*/\*/\*/ is matched by the pattern but is a “bad” comment.

5. {SLASH}{STAR}[  $\wedge$  \*] \*{STAR}+{SLASH}+ is incorrect pattern because the pattern allows the comment end with more than one slash.

/\* // \*\*\*// is matched by the pattern but is a “bad” comment.

6. {SLASH}{STAR}([  $\wedge$  \*] | ({STAR}+[  $\wedge$  \*/])) \*{STAR}+{SLASH} is incorrect pattern because the pattern allows the comment body to include \*/.

/\* /// \*\*/ \*/ is matched by the pattern but is a “bad” comment.

Question3:

```
import java_cup.runtime.*; // defines the Symbol class

class Count{
    static int num;
}

%%

DIGIT=      [0-9]
LETTER=     [a-zA-Z]
WHITESPACE= [ \t\n]      // space, tab, newline

// with java CUP (the Java parser generator)
%implements java_cup.runtime.Scanner
%function next_token
%type java_cup.runtime.Symbol

%eofval{
    System.out.println(Count.num);
    return new Symbol(sym.EOF);
%eofval}

// Turn on line counting
%line

%%

^ myFunc(((([a-z]|_)([0-9]|[a-z]|_)*)?),(([a-z]|_)([0-9]|[a-z]|_)*)*\} {
    Count.num++;
    System.out.println(yyline);
}
```