

## NAME

Parsers::SimpleCalcYYLexer

## SYNOPSIS

```
use Parsers::SimpleCalcYYLexer;

use Parsers::SimpleCalcYYLexer qw(:all);
```

## DESCRIPTION

SimpleCalcYYLexer class provides the following methods:

new, YYLex, GetYYLex, StringifySimpleCalcYYLexer

Parser::SimpleCalcYYLexer class is derived from Parser::YYLexer class, which in turn is derived from base class Parser::Lexer that provides all the underlying lexer functionality. SimpleCalcYYLexer class is designed to be used with yyparse code generated by running byacc on a parser defined using parser definition SimpleCalcParser.yy file.

The parser package and token table files, SimpleCalcParser.pm and SimpleCalcParser.tab.ph, are automatically generated from parser grammar definition file, SimpleCalcParser.yy, using byacc available through perl-byacc1.8 modified with perl5-byacc-patches-0.5 for generation of object oriented parser:

```
byacc -l -P -d -b SimpleCalcParser SimpleCalcParser.yy
mv SimpleCalcParser.tab.pl SimpleCalcParser.pm
```

SimpleCalcYYLexer.pm class implements a lexer for a simple calculator and is provided to highlight usage of YYLex through yyparse.

The default specification of lexer tokens for SimpleCalcYYLexer.pm includes:

```
@YYLexerTokensSpec = (
    [ 'LETTER', qr/[a-zA-Z]/ ],
    [ 'NUMBER', qr/[d+]/ ],
    [ 'SPACE', qr/[ ]*/ ],
    sub { my($This, $TokenLabel, $MatchedText) = @_; return ''; }
],
    [ 'NEWLINE', qr/(?:\r\n|\r|\n)/ ],
    sub { my($This, $TokenLabel, $MatchedText) = @_; return "\n"; }
],
    [ 'CHAR', qr/[. / ]
);
```

The default SimpleCalcParser.tab.ph file containing token identifiers for SimpleCalcParser.yy includes:

```
$NUMBER=257;
$LETTER=258;
```

## METHODS

new

```
$SimpleCalcYYLexer = new Parsers::SimpleCalcYYLexer($Input);
```

Using specified *Input*, new method generates a new SimpleCalcYYLexer and returns a reference to newly created SimpleCalcYYLexer object.

Examples:

```
# Input string...
$InputText = "3 + 4 +6\nx=3\nny=5\nx+y\nx+z\n";

$YYLexer = new Parsers::SimpleCalcYYLexer($InputText);
$YYLex = $YYLexer->GetYLex();

$Debug = 0;
```

```
$CalcParser = new Parsers::SimpleCalcParser($YYLex,  
                                             \&Parsers::SimpleCalcParser::yyerror, $Debug);  
$Value = $SimpleCalcParser->yyparse();  
print "Value: $Value\n";  
  
# Input file...  
$InputFile = "Input.txt";  
open INPUTFILE, "$InputFile" or die "Couldn't open $InputFile: $!\n";  
$YYLexer = new Parsers::SimpleCalcYYLexer($InputFile);  
$YYLex = $YYLexer->GetYYLex();  
  
$CalcParser = new Parsers::SimpleCalcParser($YYLex,  
                                             \&Parsers::SimpleCalcParser::yyerror);  
$Value = $SimpleCalcParser->yyparse();  
print "Value: $Value\n";  
  
# Input file iterator...  
$InputFile = "TestSimpleCalcParser.txt";  
open INPUTFILE, "$InputFile" or die "Couldn't open $InputFile: $!\n";  
$InputIterator = sub { return <INPUTFILE>; };  
$YYLexer = new Parsers::SimpleCalcYYLexer($InputIterator);  
$YYLex = $YYLexer->GetYYLex();  
  
$CalcParser = new Parsers::SimpleCalcParser($YYLex,  
                                             \&Parsers::SimpleCalcParser::yyerror);  
$Value = $SimpleCalcParser->yyparse();  
print "Value: $Value\n";
```

#### StringifySimpleCalcYYLexer

```
$YYLexerString = $YYLexer->StringifySimpleCalcYYLexer();
```

Returns a string containing information about *YYLexer* object.

#### AUTHOR

Manish Sud <msud@san.rr.com>

#### SEE ALSO

Lexer.pm, YYLexer.pm, SimpleCalcParser.yy

#### COPYRIGHT

Copyright (C) 2025 Manish Sud. All rights reserved.

This file is part of MayaChemTools.

MayaChemTools is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or (at your option) any later version.