

CS335 Group 11

Abhay Mishra (190017), Ashok Kumar Saini (190195),
Ashutosh Shukla (170171), Gagan Aryan (190327)

February 1, 2022

Repository

Github: <https://github.com/gagan-aryan/cs335-compiler-design>

Structure

```
.
|-- docs
|-- |-- CS335_Milestone1.pdf
|-- Makefile
|-- README.md
|-- src
|-- |-- lex_c.l
|-- |-- lex.l
|-- |-- lex.yy.cc
|-- |-- y.tab.h
|-- test
|-- |-- run_test.sh
|-- |-- lex
|-- |-- |-- constants.go
|-- |-- |-- for.go
|-- |-- |-- hello.go
|-- |-- |-- if-else.go
|-- |-- |-- variable.go
```

4 directories, 13 files

Output of git show lexer

```
commit 0b6e521bd74e409b7b5cb11cc99ae11e8e0f9b3e
Author: Abhay Mishra <mishraabhay9179913420@gmail.com>
Date: Tue Feb 1 23:34:00 2022 +0530
```

add test script and update readme

```
diff --git a/.clang-format b/.clang-format
new file mode 100644
index 0000000..9134b3a
--- /dev/null
+++ b/.clang-format
@@ -0,0 +1,206 @@
+---
+Language: Cpp
+BasedOnStyle: Google
+IndentWidth: 4
+# AccessModifierOffset: -1
+# AlignAfterOpenBracket: Align
+# AlignConsecutiveMacros: None
```

```

+# AlignConsecutiveAssignments: None
+# AlignConsecutiveBitFields: None
+# AlignConsecutiveDeclarations: None
+# AlignEscapedNewlines: Left
+# AlignOperands: Align
+# AlignTrailingComments: true
+# AllowAllArgumentsOnNextLine: true
+# AllowAllConstructorInitializersOnNextLine: true
+# AllowAllParametersOfDeclarationOnNextLine: true
+# AllowShortEnumsOnASingleLine: true
+# AllowShortBlocksOnASingleLine: Never
+# AllowShortCaseLabelsOnASingleLine: false
+# AllowShortFunctionsOnASingleLine: All
+# AllowShortLambdasOnASingleLine: All
+# AllowShortIfStatementsOnASingleLine: WithoutElse
+# AllowShortLoopsOnASingleLine: true
+# AlwaysBreakAfterDefinitionReturnType: None
+# AlwaysBreakAfterReturnType: None
+# AlwaysBreakBeforeMultilineStrings: true
+# AlwaysBreakTemplateDeclarations: Yes
+# AttributeMacros:
+#   - __capability
+# BinPackArguments: true
+# BinPackParameters: true
+# BraceWrapping:
+#   AfterCaseLabel: false
+#   AfterClass: false
+#   AfterControlStatement: Never
+#   AfterEnum: false
+#   AfterFunction: false
+#   AfterNamespace: false
+#   AfterObjCDeclaration: false
+#   AfterStruct: false
+#   AfterUnion: false
+#   AfterExternBlock: false
+#   BeforeCatch: false
+#   BeforeElse: false
+#   BeforeLambdaBody: false
+#   BeforeWhile: false
+#   IndentBraces: false
+#   SplitEmptyFunction: true
+#   SplitEmptyRecord: true
+#   SplitEmptyNamespace: true
+# BreakBeforeBinaryOperators: None
+# BreakBeforeConceptDeclarations: true
+# BreakBeforeBraces: Attach
+# BreakBeforeInheritanceComma: false
+# BreakInheritanceList: BeforeColon
+# BreakBeforeTernaryOperators: true
+# BreakConstructorInitializersBeforeComma: false
+# BreakConstructorInitializers: BeforeColon
+# BreakAfterJavaFieldAnnotations: false
+# BreakStringLiterals: true
+# ColumnLimit: 80
+# CommentPragmas: '~ IWYU pragma:'
+# CompactNamespaces: false
+# ConstructorInitializerAllOnOneLineOrOnePerLine: true
+# ConstructorInitializerIndentWidth: 4
+# ContinuationIndentWidth: 4
+# Cpp11BracedListStyle: true
+# DeriveLineEnding: true
+# DerivePointerAlignment: true

```

```

+# DisableFormat: false
+# EmptyLineAfterAccessModifier: Never
+# EmptyLineBeforeAccessModifier: LogicalBlock
+# ExperimentalAutoDetectBinPacking: false
+# FixNamespaceComments: true
+# ForEachMacros:
+#   - foreach
+#   - Q_FOREACH
+#   - BOOST_FOREACH
+# IncludeBlocks: Regroup
+# IncludeCategories:
+#   - Regex:      '^<ext/.*\.h>'
+#     Priority:    2
+#     SortPriority: 0
+#     CaseSensitive: false
+#   - Regex:      '^<.*\.h>'
+#     Priority:    1
+#     SortPriority: 0
+#     CaseSensitive: false
+#   - Regex:      '^<.*'
+#     Priority:    2
+#     SortPriority: 0
+#     CaseSensitive: false
+#   - Regex:      '.*'
+#     Priority:    3
+#     SortPriority: 0
+#     CaseSensitive: false
+# IncludeIsMainRegex: '([-_](test|unittest))?$'
+# IncludeIsMainSourceRegex: ''
+# IndentAccessModifiers: false
+# IndentCaseLabels: true
+# IndentCaseBlocks: false
+# IndentGotoLabels: true
+# IndentPPDirectives: None
+# IndentExternBlock: AfterExternBlock
+# IndentRequires: false
+# IndentWrappedFunctionNames: false
+# InsertTrailingCommas: None
+# JavaScriptQuotes: Leave
+# JavaScriptWrapImports: true
+# KeepEmptyLinesAtTheStartOfBlocks: false
+# MacroBlockBegin: ''
+# MacroBlockEnd: ''
+# MaxEmptyLinesToKeep: 1
+# NamespaceIndentation: None
+# ObjCBinPackProtocolList: Never
+# ObjCBlockIndentWidth: 2
+# ObjCBreakBeforeNestedBlockParam: true
+# ObjCSPACEAfterProperty: false
+# ObjCSPACEBeforeProtocolList: true
+# PenaltyBreakAssignment: 2
+# PenaltyBreakBeforeFirstCallParameter: 1
+# PenaltyBreakComment: 300
+# PenaltyBreakFirstLessLess: 120
+# PenaltyBreakString: 1000
+# PenaltyBreakTemplateDeclaration: 10
+# PenaltyExcessCharacter: 1000000
+# PenaltyReturnTypeOnItsOwnLine: 200
+# PenaltyIndentedWhitespace: 0
+# PointerAlignment: Left
+# RawStringFormats:
+#   - Language: Cpp

```

```

+# Delimiters:
+#   - cc
+#   - CC
+#   - cpp
+#   - Cpp
+#   - CPP
+#   - 'c++'
+#   - 'C++'
+# CanonicalDelimiter: ''
+# BasedOnStyle:    google
+# - Language:      TextProto
+# Delimiters:
+#   - pb
+#   - PB
+#   - proto
+#   - PROTO
+# EnclosingFunctions:
+#   - EqualsProto
+#   - EquivToProto
+#   - PARSE_PARTIAL_TEXT_PROTO
+#   - PARSE_TEST_PROTO
+#   - PARSE_TEXT_PROTO
+#   - ParseTextOrDie
+#   - ParseTextProtoOrDie
+#   - ParseTestProto
+#   - ParsePartialTestProto
+# CanonicalDelimiter: pb
+# BasedOnStyle:    google
+# ReflowComments:  true
+# ShortNamespaceLines: 1
+# SortIncludes:     CaseSensitive
+# SortJavaStaticImport: Before
+# SortUsingDeclarations: true
+# SpaceAfterCStyleCast: false
+# SpaceAfterLogicalNot: false
+# SpaceAfterTemplateKeyword: true
+# SpaceBeforeAssignmentOperators: true
+# SpaceBeforeCaseColon: false
+# SpaceBeforeCpp11BracedList: false
+# SpaceBeforeCtorInitializerColon: true
+# SpaceBeforeInheritanceColon: true
+# SpaceBeforeParens: ControlStatements
+# SpaceAroundPointerQualifiers: Default
+# SpaceBeforeRangeBasedForLoopColon: true
+# SpaceInEmptyBlock: false
+# SpaceInEmptyParentheses: false
+# SpacesBeforeTrailingComments: 2
+# SpacesInAngles:  Never
+# SpacesInConditionalStatement: false
+# SpacesInContainerLiterals: true
+# SpacesInCStyleCastParentheses: false
+# SpacesInLineCommentPrefix:
+#   Minimum:        1
+#   Maximum:         -1
+# SpacesInParentheses: false
+# SpacesInSquareBrackets: false
+# SpaceBeforeSquareBrackets: false
+# BitFieldColonSpacing: Both
+# Standard:         Auto
+# StatementAttributeLikeMacros:
+#   - Q_EMIT
+# StatementMacros:

```

```

+# - Q_UNUSED
+# - QT_REQUIRE_VERSION
+# TabWidth:      8
+# UseCRLF:       false
+# UseTab:        Never
+# WhitespaceSensitiveMacros:
+# - STRINGIZE
+# - PP_STRINGIZE
+# - BOOST_PP_STRINGIZE
+# - NS_SWIFT_NAME
+# - CF_SWIFT_NAME
+# ...
+
diff --git a/.gitignore b/.gitignore
index e86ee73..de74700 100644
--- a/.gitignore
+++ b/.gitignore
@@ -1,3 +1,2 @@
-.clang-format
*.out
*.o
diff --git a/Makefile b/Makefile
index 17ea22c..251522d 100644
--- a/Makefile
+++ b/Makefile
@@ -1,10 +1,21 @@
-# Format all cpp file using clang-format before compiling
-all: build
+SOURCE_DIR="./src"
+TEST_DIR="./test"

-buildc: src/lex_c.l
-flex -o src/lex.yy.c src/lex_c.l
-gcc src/lex.yy.c -lfl -o src/lex_c.o
+all: build

build: src/lex.l
flex -+ -o src/lex.yy.cc src/lex.l
g++ src/lex.yy.cc -lfl -o src/lex.o
+
+clean:
+ rm -f $(SOURCE_DIR)/*.o
+ rm -rf $(TEST_DIR)/lex/output
+
+# TODOs:
+# 1. Automate testing: write input, output and compare automatically
+#      ( ? Use github actions )
+# 2. Format files: Avoids unnecessary changes. Use:
+#      a) TERMINAL: clang-format -i FILE_TO_FORMAT
+#      b) VSCODE: Install clang-format extension
+#                  Configure default formatter as clang-format
+#                  Ctrl + Shift + I
\ No newline at end of file
diff --git a/README.md b/README.md
index 80ed3b4..6a742eb 100644
--- a/README.md
+++ b/README.md
@@ -1,2 +1,15 @@
# cs335--compiler-design
-Project for the course CS335, Spring 2022
+Project for the course CS335, Spring 2022
+

```

```

+## Milestone 2 - Lexical Analysis
+
+1. Building
+Makefile is provided. Simply run 'make' in top level directory.
+Use 'make clean' as needed.
+
+2. Running test cases
+'''
+cd test
+./run_test.sh lex
+'''
+This would generate the outputs in 'lex/output' directory.
\ No newline at end of file
diff --git a/fmt b/fmt
deleted file mode 100644
index e69de29..0000000
diff --git a/src/lex.l b/src/lex.l
index a936b41..ff0798f 100755
--- a/src/lex.l
+++ b/src/lex.l
@@ -50,7 +50,7 @@ newline [\n]
 {letter}({letter}|{dec_digit})*      { //yylval.id = strdup(lexer->YYText());
                                     return ID; }

-{dec_digit}+ {return CONST;}
+{dec_digit}+ {return DEC_LIT;}
(0)[bB]{binary_digit}+ {return BINARY_LIT;}
(0)[xX]{hex_digit}+ {return HEX_LIT;}
{dec_digit}*"."{dec_digit}|{dec_digit}*"."{dec_digit}*[eE][+-]?{dec_digit}*|{dec_digit}*[eE][+-]
]
@@ -204,8 +204,10 @@ int main()
    case 1060:
        cout<<"FLOAT_LIT"<<"\t"<<lexer->YYText() <<"\t"<<row << "\t" << column<<"\n";
        break;
+    case 1061:
+        cout<<"DEC_LIT"<<"\t"<<lexer->YYText() <<"\t"<<row << "\t" << column<<"\n";
+        break;
    }
    column+=lexer->YYLeng();
}
- printf("BYE\n");
}
diff --git a/src/lex.yy.cc b/src/lex.yy.cc
index 6862c79..32356ef 100644
--- a/src/lex.yy.cc
+++ b/src/lex.yy.cc
@@ -853,7 +853,7 @@ YY_RULE_SETUP
    case 31:
        YY_RULE_SETUP
        #line 53 "src/lex.l"
-{return CONST;}
+{return DEC_LIT;}
        YY_BREAK
    case 32:
        YY_RULE_SETUP
@@ -2031,9 +2031,11 @@ int main()
    case 1060:
        cout<<"FLOAT_LIT"<<"\t"<<lexer->YYText() <<"\t"<<row << "\t" << column<<"\n";
        break;
+    case 1061:
+        cout<<"DEC_LIT"<<"\t"<<lexer->YYText() <<"\t"<<row << "\t" << column<<"\n";
+        break;
    }
}

```

```

        column+=lexer->YYLeng();
    }
-   printf("BYE\n");
}

diff --git a/src/y.tab.h b/src/y.tab.h
index 863a84d..9b519d6 100755
--- a/src/y.tab.h
+++ b/src/y.tab.h
@@ -36,6 +36,7 @@ enum KEYWORD {
    STRING_LIT,
    BINARY_LIT,
    HEX_LIT,
-   FLOAT_LIT
+   FLOAT_LIT,
+   DEC_LIT
};

diff --git a/test/lex/variable.go b/test/lex/variable.go
index f9ab285..164f0b4 100644
--- a/test/lex/variable.go
+++ b/test/lex/variable.go
@@ -13,9 +13,6 @@ func main() {
    var d = true
    fmt.Println(d)

-   var e int
-   fmt.Println(e)
-
    f = "apple"
    fmt.Println(f)
}
\ No newline at end of file
diff --git a/test/run_test.sh b/test/run_test.sh
new file mode 100755
index 0000000..f6e3ec3
--- /dev/null
+++ b/test/run_test.sh
@@ -0,0 +1,35 @@
+#!/bin/bash
+
+# Usage: To run "lex" tests:
+#       cd test
+#       ./run_test.sh lex
+
+PROJECT_HOME=${PWD}/..
+SOURCE_DIR=${PROJECT_HOME}/src
+
+# lex | syntax | ..
+TEST_CLASS=$1
+
+TEST_DIR=${PROJECT_HOME}/test/${TEST_CLASS}
+if [ ! -d ${TEST_DIR} ]
+then
+   echo "No tests present for ${TEST_CLASS}"
+   exit
+fi
+
+TEST_OUTPUT_DIR=${TEST_DIR}/output
+if [ ! -d ${TEST_OUTPUT_DIR} ]
+then
+   mkdir ${TEST_OUTPUT_DIR}

```

```
+fi
+
+# Name of test directory and source file should be same
+# for this to work. e.g. lex.o and test/lex
+for TEST_FILE in ${TEST_DIR}/*
+do
+    if [ -d ${TEST_FILE} ]; then continue; fi;
+    # constant.go -> constant.output.txt
+    TEST_OUTPUT_FILE=${TEST_OUTPUT_DIR}/${basename -a $TEST_FILE}.output.txt
+    rm -f ${TEST_OUTPUT_FILE}
+    ${SOURCE_DIR}/${TEST_CLASS}.o < ${TEST_FILE} > ${TEST_OUTPUT_FILE}
+done
```