

MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2270582 COMPILER DESIGN LAB

B. Tech.IV Year-I Sem

L / T / P / C 0 / 0 / 2 / 1

COURSE OUTCOMES - CO'S

- Apply lexical analysis techniques to recognize **identifiers and reserved keywords** in source code using tools like **LEX**, reinforcing understanding of scanning and token generation.
- Implement **top-down parsing algorithms**, such as **Predictive Parsing**, to analyze the syntactic structure of a given input based on its grammar.
- Construct **three-address code generators** for translating high-level language constructs into intermediate representations used in compilers.
- Design and implement **bottom-up parsers** using **SLR(1)** and **LALR(1)** techniques to handle more complex grammar structures efficiently and accurately.
- Develop and simulate parsing of a given programming language grammar (with constructs like loops, conditionals, arrays, and scoping) and handle comments and nested blocks using custom-designed parsers.

LIST OF EXPERIMENTS:

- 1. Design and develop interactive and dynamic web applications using HTML, CSS, JavaScript and XML
- 2. Apply client-server principles to develop scalable and enterprise web applications.
- 3. Ability to design, develop, and implement a compiler for any language.
- 4. Able to use lex and yacc tools for developing a scanner and a parser.
- 5. Able to design and implement LL and LR parsers.

<program> ::= <block>

```
<block> ::= { <variabledefinition> <slist> }
```

```
| { <slist> }
```

<variabledefinition> ::= int <vardeflist> ;

```
<vardeflist> ::= <vardec> | <vardec> , <vardeflist>
```

```
<vardec> ::= <identifier> | <identifier> [ <constant> ]
```

```
<slist> ::= <statement> | <statement> ; <slist>
```

<statement> ::= <assignment> | <ifstatement> | <whilestatement>

| <block> | <printstatement> | <empty>



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT (AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad) Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act, 1956

```
<assignment> ::= <identifier> = <expression>
                                          | <identifier> [ <expression> ] = <expression>
             <ifstatement> ::= if <bexpression> then <slist> else <slist> endif
                                           | if <bexpression> then <slist> endif
<whilestatement> ::= while <bexpression> do <slist> enddo
<printstatement> ::= print ( <expression> )
<expression> ::= <expression> <addingop> <term> | <term> | <addingop> <term>
<br/>

<relop> ::= < | <= | == | >= | > | !=
<addingop> ::= + | -
<term> ::= <term> <multop> <factor> | <factor>
<multop> ::= * | /
<factor> ::= <constant> | <identifier> | <identifier> | <constant> |
           ( <expression> )
<constant> ::= <digit> | <digit> <constant>
<identifier> ::= <identifier> <letterordigit> | <letter>
<letterordigit> ::= <letter> | <digit>
\langle | etter \rangle ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z
<digit> ::= 0|1|2|3|4|5|6|7|8|9
<empty> has the obvious meaning
Comments (zero or more characters enclosed between the standard C/Java-style comment brackets
```

/*...*/) can be inserted. The language has rudimentary support for 1-dimensional arrays. The declaration int a[3] declares an array of three elements, referenced as a[0], a[1] and a[2]. Note also that you should worry about the scoping of names.

A simple program written in this language is:

```
{ int a[3],t1,t2;t1=2;
a[0]=1; a[1]=2; a[t1]=3;
t2=-(a[2]+t1*6)/(a[2]-t1);
if t2>5 then print(t2); else
{
    int t3; t3=99; t2=-25;
    print(-t1+t2*t3); /* this is a comment
```