CBCS SCHEME

USN | | | | | | | | | | |

18CS61

Sixth Semester B.E. Degree Examination, June/ July 2025
System Software and Compilers

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Explain SIC architecture. **(10 Marks)**

b. Generate the complete object program for the following SIC /XE code.

```
SUM  START  0
FIRST LDX # 0
        LDA # 0
       + LDB # TABLE 2
        BASE TABLE2
LOOP ADD TABLE, X
       ADD TABLE 2, X
       TIX COUNT
       JLT LOOP
      + STA TOTAL
       RSUB
 COUNT RESW1
TABLE RESW 2000
TABLE2  RESW 2000
TOTAL    RESW1
           END FIRST
```

OP codes:

ADD – 18, JLT – 38, LDA – 00, LDB -68, LDX – 04, RSUB – 4C, TIX – 24, STA – 1C **(10 Marks)**

OR

2 a. Write the algorithms of Absolute loader and Boot strap loader. **(10 Marks)**

b. List the machine independent features of assembler. Explain literals and program blocks. **(10 Marks)**

Module-2

3 a. Explain the different phases of the compiler and show the translation of the statement. Position = initial + rate * 60. **(10 Marks)**

b. Determine the role of sentinels in input buffering along with the look ahead code. **(10 Marks)**

OR

4 a. Write the regular definitions and transition diagrams of relational operators, identifiers, unsigned numbers and white spaces. **(10 Marks)**

b. Write the code to generate the tokens for relational operators. **(04 Marks)**

c. Define Token, pattern and lexeme with examples. **(06 Marks)**

Module-3

5 a. Explain the FIRST and FOLLOW rules used in predictive parsing technique and identify the FIRST and FOLLOW SETS in the following grammar.

$E \rightarrow TE'$

$E' \rightarrow TE' \mid E$

$T \rightarrow FT'$

$T' \rightarrow * FT' \mid E$

$F \rightarrow (E) \mid id$ **(10 Marks)**

b. $S \rightarrow (L) \mid a$

$L \rightarrow L, s \mid s$

Modify the grammar suitable for Top-Down Parsing, construct LL(1) parsing table and show the parsing moves for the string (a,a) **(10 Marks)**

OR

6 a. Write the algorithm for Left factoring a grammar and ; illustrate left factoring for the grammar

$S \rightarrow iEts \mid iEtses \mid a$

$E \rightarrow b$ **(10 Marks)**

b. Explain the actions and conflicts of shift reduce parser. Show the parsing moves for the string $id_1 * id_2$ considering the grammar

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow id$ **(10 Marks)**

Module-4

7 a. Write the lexer with symbol table for recognizing parts of speech. **(10 Marks)**

b. Explain the different sections of the YACC parser and also write a simple lexer to provide tokens to the parser using a simple example program. **(10 Marks)**

OR

8 a. Explain the rules section of the extended English parser for recognizing simple and compound sentence. **(10 Marks)**

b. Explain any ten characters that form regular expressions in defining the patterns in Lex code. **(10 Marks)**

Module-5

9 a. Explain syntax directed definition and write the SDD for simple desk calculator, annotated parse tree and dependency graph for 6+ 5 * 7 n. **(10 Marks)**

b. Construct and explain quadruples, triples and Indirect triples for the statement a + (-b * c) + (c * d). **(10 Marks)**

OR

10 a. Construct DAG for the following expression along with SDD and steps for the expression a + a * (b – c) + (b – c) * d **(10 Marks)**

b. Identify and explain the issues in the design of code generator. **(10 Marks)**