

CBCS SCHEME

USN

--	--	--	--	--	--	--	--	--	--

15CS63

Sixth Semester B.E. Degree Examination, June/July 2019 System Software and Compiler Design

Time: 3 hrs.

Max. Marks: 80

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

Module-1

- 1 a. Explain SIC/XE architecture. (08 Marks)
b. Generate the complete object program for the following SIC/XE assembly program.

```
WRREC  START  405D
        CLEAR  X
        LDT    LENGTH
WLOOP  TD      OUTPUT
        JEQ    WLOOP
        LDCH  BUFFER, X
        WD    OUTPUT
        TIXR  T
        JLT   WLOOP
        RSUB
OUTPUT BYTE  X '05'
        END
Address of BUFFER      4033
Address of LENGTH     4036
```

Op Codes :

CLEAR – B4 ; JEQ – 30; WD – DC; JLT – 38;
LDT – 74; LDCH – 50; TIXR – B8; RSUB – 4C. (08 Marks)

OR

- 2 a. List all assembler independent and dependant features and explain program relocation. (05 Marks)
b. Explain the data structures used in macro processor with example. (03 Marks)
c. Explain the following macroprocessor independent features. (08 Marks)
i) Generation of unique lables
ii) Keyword macro parameter.

Module-2

- 3 a. What is loader? What are the basic functions the loader has to perform? (04 marks)
b. Develop an algorithm for bootstrap loader. (07 marks)
c. Explain dynamic linking with suitable diagram. (05 Marks)

OR

- 4 a. Differentiate between a linking loader and linkage editor, with the help of suitable diagram. (08 marks)
b. Explain different loader option commands with examples. (04 marks)
c. Illustrate MS – DOS object module with its record types. (04 Marks)

Module-3

- 5 a. With the help of a diagram, explain the various phases of compiler. (08 Marks)
 b. Explain the concept of input buffering in the lexical analysis. (04 Marks)
 c. What design objectives, compiler optimizations must meet. (04 Marks)

OR

- 6 a. Write a LEX program for the tokens given below : (08 Marks)

LEXEMES	TOKEN NAME	ATTRIBUTE VALUE
Any WS	—	—
if	if	—
then	then	—
else	else	—
Any id	id	ptr to table entry
Any number	number	ptr to table entry
<	reloop	LT
<=	reloop	LE
=	reloop	EQ
< >	reloop	NE
>	reloop	GT
>=	reloop	GE

- b. Write regular definitions for unsigned numbers and draw the transition diagram for the same. (08 Marks)

Module-4

- 7 a. Define left recursion grammer, eliminate left recursion from the following grammer :
 $S \rightarrow aB \mid ac \mid sd \mid se$
 $B \rightarrow b Bc \mid f$
 $C \rightarrow g$ (03 Marks)
- b. Consider the following context free grammer $S \rightarrow SS + \mid SS * \mid a$ and the input string $aa + a*$
 i) Give LMD and RMD
 ii) Parse tree
 iii) Is the grammer ambiguous? Why
 iv) Describe the language generated by the grammer
 v) Left factor the grammer. (05 Marks)
- c. Consider the following grammer with terminals (, [,) ,]
 $S \rightarrow TS \mid [S] S \mid)S \mid \epsilon$
 $T \rightarrow (x)$
 $X \rightarrow TX \mid [X] X \mid \epsilon$
 i) Construct first and follow sets
 ii) Construct its LL(1) parsing table
 iii) Is this grammer LL(1)? (08 marks)

OR

- 8 a. The following is ambiguous grammar

$$S \rightarrow AS \mid b$$

$$A \rightarrow SA \mid a$$

Construct for this grammar its collection of sets of LR(0) items. If we try to build an LR – parsing table for the grammar, there are certain conflicting actions what are they? Suppose we tried to use the parsing table by non deterministically choosing a possible action whenever there is a conflict, show all the possible sequences of actions on input abab\$.

(10 Marks)

- b. What are the actions of a shift – reduce parser. Design shift – reduce parser for the following grammar on the input 10201 $S \rightarrow 0 S 0 \mid 1 S 1 \mid 2$.

(06 Marks)

Module-5

- 9 a. Consider the context free grammar given below :

$$S \rightarrow EN$$

$$E \rightarrow E + T \mid E - T \mid T$$

$$T \rightarrow T * F \mid T / F \mid F$$

$$F \rightarrow (E) \mid \text{digit}$$

$$N \rightarrow ;$$

i) Obtain the SDD for the above grammar

ii) Construct annotated parse tree for the input string $5 * 6 + 7$.

(08 Marks)

- b. Obtain the DAG for the expression, show the steps
- $a + a * (b - c) + (b - c) * d$
- .

(04 Marks)

- c. Translate the assignment

$$a = b * - c + b * - c$$
 into

i) Three address code

ii) Quadruples.

(04 Marks)

OR

- 10 a. Explain the issues in the design of a code generator.

(11 marks)

- b. Write the machine instructions for the following three address instructions :

i) $b = a[i]$ ii) $a[j] = c$ iii) $x = *p$ iv) $*p = y$ v) if $x < y$ got L.

(05 Marks)
