CBCS SCHEME

USN | | | | | | | | | | 15CS63

Sixth Semester B.E. Degree Examination, Dec.2018/Jan.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 in detail SIC/XE machine architecture. **(08 Marks)**

b. Write an SIC/XE program to calculate DELTA = ALPHA + BETA * GAMMA – 10 **(08 Marks)**

OR

2 a. Write an algorithm for Pass – 1 of an assembler. **(08 Marks)**

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

```
SUM     START   0
FIRST   CLEAR   X
        LDA     #0
        +LDB    #TOTAL
        BASE    TOTAL
LOOP    ADD     TABLE, X
        TIX     COUNT
        JLT     LOOP
        STA     TOTAL
COUNT   RESW    1
TABLE   RESW    2000
TOTAL   RESW    1
        END     FIRST
```

Mnemonic	ADD	JLT	LDA	LDB	LDX	RSUB	STA	TIX	JSUB	J	LDT	CLEAR
opcode	18	38	00	68	04	4C	0C	2C	08	3C	74	B4

(08 Marks)

Module-2

3 a. Write PASS-1 and PASS-2 algorithm for a linking loader. **(08 Marks)**

b. Explain dynamic linking, automatic library search, loader design options with suitable examples. **(08 Marks)**

OR

4 a. Write the SIC/XE program for a bootstrap loader with suitable comments. Explain in brief the algorithm of a bootstrap loader. **(08 Marks)**

b. Explain in brief (i) MS-DOS linker and (ii) CRAY MPP linker. **(08 Marks)**

Module-3

5 a. List and explain the various phases of a compiler and show the output of each phase for the expression a := b + c * 25 **(08 Marks)**

b. Construct transition diagram for recognizing relational operators. Sketch the program segment to implement it, showing the first state and one in final state. **(08 Marks)**

OR

6 a. Explain input buffering strategy used in lexical analysis phase. **(06 Marks)**
b. Write the regular definition for unsigned number, also write the transition diagram. **(06 Marks)**
c. Construct the transition diagrams for a set of keywords like begin, end, if then and else and, identifiers and constants along with a minimum set of relational operators. **(04 Marks)**

Module-4

7 a. What is shift reduce parser? Explain the conflicts that may occur during shift reduce parsing. **(04 Marks)**
b. Construct LALR parsing tables for the grammar shown below using LR(1) items.
$S' \rightarrow S$
$S \rightarrow Cc$
$C \rightarrow cC \mid d$ **(08 Marks)**
c. How left recursion can be eliminated from grammars? Write down the simple arithmetic expression grammar and rewrite the grammar after removing left recursion. **(04 Marks)**

OR

8 a. What is left factoring? Rewrite the following grammar after "left factored"
$S \rightarrow iEts \mid iEtSeS \mid a$
$E \rightarrow b$ **(04 Marks)**
b. Write a note on the parser generator – yacc. **(04 Marks)**
c. Construct canonical LR(1) items for the augmented grammar
$S' \rightarrow S$:
$S \rightarrow Cc$
$C \rightarrow cC \mid d$ **(08 Marks)**

Module-5

9 a. Define synthesized and inherited attributes with examples. **(04 Marks)**
b. Briefly explain the main issues in code generation. **(08 Marks)**
c. Explain in brief dead code elimination. **(04 Marks)**

OR

10 a. Construct DAG for the expression
$a + b * (a + b) + c + d$ **(04 Marks)**
b. Give SDD of a simple calculator. **(04 Marks)**
c. Write a note on common sub expression. **(04 Marks)**
d. What are the steps involved in optimization of basic blocks. Explain any 2 steps in brief. **(04 Marks)**

* * * * *

CBCS SCHEME

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

15CS63

Sixth Semester B.E. Degree Examination, Dec.2019/Jan.2020
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 the various instruction formats used in SIC/XE machine. **(04 Marks)**

b. Write a SIC/XE program to copy the string "COMPUTER SCIENCE ENGINEERING" from STR1 to another string STR2. **(06 Marks)**

c. List the functions of Pass-1 and Pass-2 of a two pass assembler. **(06 Marks)**

OR

2 a. Write an algorithm of the Pass-1 of a two pass assembler. **(08 Marks)**

b. List the various machine independent assembler features. Explain the control-sections, how the assembler converter them into object code. **(08 Marks)**

Module-2

3 a. Define Macro. Explain how Macros are defined and expanded. **(07 Marks)**

b. What are the basic functions of a loader? Explain two ways of program relocation in loaders. **(09 Marks)**

OR

4 a. Explain the functions of dynamic linking with a diagram. **(08 Marks)**

b. Write a note on MS-DOS linker. **(08 Marks)**

Module-3

5 a. Explain the different phases of a compiler, with an example. **(09 Marks)**

b. What is input buffering in lexical analysis? List the different methods of input buffering explain any one of them. **(07 Marks)**

OR

6 a. List and explain the reasons for separating the analysis portion of a compiler into lexical and syntax analysis phases. **(06 Marks)**

b. Construct the transition diagram to recognize the tokens of
i) Identifier ii) Relational operators iii) Unsigned numbers. **(06 Marks)**

c. Define Tokens, patterns, lexemes. **(04 Marks)**

Module-4

7 a. What is the role of parser? Explain the different error recovery strategies. **(08 Marks)**

b. Construct the LL(1) parsing table for the following productions:
$E \rightarrow E + T/T$; $T \rightarrow T * F/F$; $F \rightarrow (E)/id$ **(08 Marks)**

OR

8 a. Using operator-precedence parsing algorithm, construct the table and parse the input string id + id * id. **(12 Marks)**

b. Define Handle, viable prefixes. **(04 Marks)**

Module-5

9 a. Discuss S-attributed and L-attributed SDD. **(06 Marks)**

b. Write 3-address code syntax tree and DAG for the expression a + a * (b – c) + (b – c) * d. **(10 Marks)**

OR

10 a. Obtain the SDD and construct annotated parse tree for the input string 6 * 5 + 3, for the grammar

$S \rightarrow EN$
$E \rightarrow E + T/T$
$T \rightarrow T * F/F$
$F \rightarrow (E)/digit$
$N \rightarrow ;$ **(10 Marks)**

b. Discuss the issues in the design of code generator. **(06 Marks)**

* * * * *

CBCS SCHEME

USN | | | | | | | | | | 15CS63

Sixth Semester B.E. Degree Examination, June/July 2019
System Software and Complier 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.
i) Generation of unique lables
ii) Keyword macro parameter. **(08 Marks)**

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 complier. **(08 Marks)**

b. Explain the concept of input buffering in the lexical analysis. **(04 Marks)**

c. What design objectives, complier 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 (, [,),]

$S \rightarrow TS \mid [S]\,S \mid)S \mid \in$

$T \rightarrow (x)$

$X \rightarrow TX \mid [X]\,X \mid \in$

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 grammer

$S \rightarrow AS \mid b$

$A \rightarrow SA \mid a$

Construct for this grammer its collection of sets of LR(0) items. IF we try to build an LR – parsing table for the grammer, 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 ababS. **(10 Marks)**

b. What are the actions of a shift – reduce parser. Design shift – reduce parser for the following grammer 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 grammer 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 grammer.
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)**

* * * * *