CBCS SCHEME

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

18CS61

Sixth Semester B.E. Degree Examination, Dec.2024/Jan.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. Define System Software. Distinguish between system software and application software. **(06 Marks)**

b. Explain SIC/XE architecture **(08 Marks)**

c. Write an algorithm for pass 2 assembler of SIC assembler. **(06 Marks)**

OR

2 a. Explain the data structure and pass 1 algorithm of SIC assembler. **(08 Marks)**

b. List all assembler independent and dependent features and explain program relocation. **(06 Marks)**

c. What is loader? What are the basic functions the loader has to perform? **(06 Marks)**

Module-2

3 a. What is compiler? Explain various phases of compiler with the help of neat diagram. **(10 Marks)**

b. Explain the concept of input buffering with sentinels in the lexical analysis. **(06 Marks)**

c. List the formal definitions of operations on languages with notations. **(04 Marks)**

OR

4 a. Write the regular definition using extended regular expression notation and also draw the transition diagram to recognize the following tokens:
(i) Identifier (ii) Unsigned **(10 Marks)**

b. Explain three types of software productivity tools. **(05 Marks)**

c. Enlist algebraic laws for regular expressions. **(05 Marks)**

Module-3

5 a. Define left-recursion grammar, also write an algorithm to eliminate left recursion from a grammar. **(05 Marks)**

b. How to verify whether grammar is LL(1) or not? Show that :

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow E$

$B \rightarrow E$

is LL(1), without constructing any table. **(10 Marks)**

c. For the grammar $A \rightarrow (A) \mid a$, construct LR(0) set of items **(05 Marks)**

OR

6 a. Explain the working of shift reduce parser. Parse the input string id ∗ id using the grammar.

$E \rightarrow E + T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid id$ **(08 Marks)**

b. With a diagram, explain the model of an LR parser. **(04 Marks)**

c. For the given grammar E → E + n/n construct parsing table of LL(1). Verify 3+4+7 and show each step of verification with reference to parsing table. **(08 Marks)**

Module-4

7 a. Explain the structure of LEX program, with an example. **(06 Marks)**

b. Write a LEX program for the tokens given below:

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
<	relop	LT
<=	relop	LE
=	relop	EQ
<>	relop	NE
>	relop	GT
>=	relop	GE

(10 Marks)

c. Write a LEX program to count the number of vowels and consonants in a given input string. **(04 Marks)**

OR

8 a. List and explain with an example the different wildcard characters used in LEX. **(08 Marks)**

b. Write a YACC program to evaluate the arithmetic expression. **(06 Marks)**

c. Explain the structure of YACC program. **(06 Marks)**

Module-5

9 a. Write annotated parse tree for 3 ∗ 5 + 4n using top down approach. Write semantic rules for each step. **(10 Marks)**

b. Define (i) Synthesized attribute (ii) Inherited attribute. **(06 Marks)**

c. Explain the concept of syntax directed definition. **(04 Marks)**

OR

10 a. Construct DAG and three address code for the following expression:

$a + a * (b - c) + (b - c) * d$ **(04 Marks)**

b. Explain the following with an example:
i) quadruples ii) triples **(08 Marks)**

c. Discuss the various issues in the design of a code generator. **(08 Marks)**

* * * * *