SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956) THINK MERIT | THINK TRANSPARENCY | THINK SASTRA	**School of Computing** **First CIA Exam – Aug 2024** **Course Code: CSE320** **Course Name: Compiler Design** **Duration: 90 minutes** **Max Marks: 50**

PART – A

ANSWER ALL THE QUESTIONS **10 X 2 = 20 Marks**

1. A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2 and T_3 over the alphabet {a, b, c}.
   T_1: a?(b|c)*a
   T_2: b?(a|c)*b
   T_3: c?(b|a)*c
   If the string "baabbccccab" is processes by the analyzer, what will be the sequence of tokens it produces?
2. Consider the following segment of python program,
```
# float n
numbers = [11, 12, 13]
for n in numbers:
    print("{n} is a number")
```
   Specify the tokens recognized by the Lexical Analyser during scanning.
3. Given the Expression grammar of a recent high-level programming language,
   E -> E * F | F + E | F
   F -> F - F | id
   Find the precedence of operators with respect to the Grammar.
4. Consider the grammar
   E → E + n | E * n | n.
   For a sentence **n + n * n**, identify the "handles" in the right-sentential form of the reduction technique?
5. Build the derivation tree for the string **((a, a), a)** based on the following Context Free grammar
   S → (L) | a
   L → L , S | S
6. Illustrate the process of eliminating the Left Recursion from the given grammar:
   A → ABb | Bb |a
   B → Bc | Acc |b

SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956) THINK MERIT | THINK TRANSPARENCY | THINK SASTRA	**School of Computing** **First CIA Exam – Aug 2024** **Course Code: CSE320** **Course Name: Compiler Design** **Duration: 90 minutes** **Max Marks: 50**

7. Write the **LEX specification** to recognize the string w described as:
   - w begins and ends with digit. |w| between 8 and 12 characters.
   - w contains atleast one Upper case letter and a special character (either # or @) in the middle.
8. Augment a Context Free Grammar to represent the syntactical correctness of **while and do-while statement in C++.**
9. Compare the behaviour of the Compiler and Interpreter upon detecting any error in the Source program during the translation.
10. Consider the language of all strings over {a, b, c} containing the substring "**abcabb**". Write a regular expression that describing this language.

PART – B

ANSWER ALL THE QUESTIONS **3 X 10 = 30 Marks**

11. Construct a LL(1) Parsing Table for the given Context Free Grammar.
    A → (B) | a
    B → B , A | A
12. Design a Lexical Analyser Generator to recognize the patterns in the following LEX specification.
```
%%
aba          {action sequence A1 for pattern P1}
ab*b         {action sequence A2 for pattern P2}
b+a+         {action sequence A3 for pattern P3}
%%
int main ( )
{ yylex();
return 0; }
int yywrap()
        { return 1;}
```
13. Show how the Deterministic Finite Automaton is converted directly from an augmented regular expression **r=(0|1)*(0|1)**

--- END---

SASTRA
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**
CO – RBT

PART – A

ANSWER ALL THE QUESTIONS **10 X 2 = 20 Marks**

No	Question	CO	RBT
1.	A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2 and T_3 over the alphabet {a, b, c}. **T_1: a?(b|c)*a** **T_2: b?(a|c)*b** **T_3: c?(b|a)*c** If the string "baabbccccab" is processes by the analyzer, what will be the sequence of tokens it produces?	1	2
2.	Consider the following segment of python program, **# float n** **numbers = [11, 12, 13]** **for n in numbers:** **print("{n} is a number")** Specify the tokens recognized by the Lexical Analyser during scanning.	1	1
3.	Given the Expression grammar of a recent high-level programming language, **E -> E * F | F + E | F** **F -> F - F | id** Find the precedence of operators with respect to the Grammar.	2	2
4.	Consider the grammar **E → E + n | E * n | n**. For a sentence **n + n * n**, identify the "handles" in the right-sentential form of the reduction technique?	2	2
5.	Build the derivation tree for the string **((a, a), a)** based on the following Context Free grammar **S → (L) | a** **L → L , S | S**	2	2
6.	Illustrate the process of eliminating the Left Recursion from the given grammar: **A → ABb | Bb |a** **B → Bc | Acc |b**	2	3

SASTRA
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**
CO – RBT

No	Question	CO	RBT
7.	Write the **LEX specification** to recognize the string w described as: ▪ w begins and ends with digit. |w| between 8 and 12 characters. ▪ w contains atleast one Upper case letter and a special character (either # or @) in the middle.	1	2
8.	Augment a Context Free Grammar to represent the syntactical correctness of **while and do-while statement in C++.**	2	2
9.	Compare the behaviour of the Compiler and Interpreter upon detecting any error in the Source program during the translation.	1	2
10.	Consider the language of all strings over {a, b, c} containing the substring "**abcabb**". Write a regular expression that describing this language.	1	2

PART – B

ANSWER ALL THE QUESTIONS **3 X 10 = 30 Marks**

No	Question	CO	RBT
11.	Construct a LL(1) Parsing Table for the given Context Free Grammar. **A → (B) | a** **B → B , A | A**	2	3
12.	Design a Lexical Analyser Generator to recognize the patterns in the following LEX specification. **%%** **aba** **{action sequence A_1 for pattern P_1}** **ab*b** **{action sequence A_2 for pattern P_2}** **b+a+** **{action sequence A_3 for pattern P_3}** **%%** int main ( ) { yylex(); return 0; } int yywrap() { return 1;}	1	3
13.	Show how the Deterministic Finite Automaton is converted directly from an augmented regular expression **r=(0|1)*(0|1)**	1	3

--- END---

SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**

ANSWER KEY

PART – A

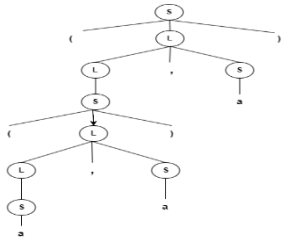
ANSWER ALL THE QUESTIONS **10 X 2 = 20 Marks**

1.	A lexical analyzer uses the following patterns to recognize three tokens T_1, T_2 and T_3 over the alphabet {a, b, c}. **T_1: a?(b|c)*a** **T_2: b?(a|c)*b** **T_3: c?(b|a)*c** If the string "baabbccccab" is processes by the analyzer, what will be the sequence of tokens it produces?
	Token T3 for the substring "baabbc" *(2marks)* *Token T2 for the remaining string "cccab"*
2.	Consider the following segment of python program, **# float n** **numbers = [11, 12, 13]** **for n in numbers:** **print("{n} is a number")** Specify the tokens recognized by the Lexical Analyser during scanning.
	Tokens are: <ID, numbers>, <SPL, [>, <LIT, 11>, <LIT, 12>, <LIT, 13>, <SPL,]> <KEY, for>, <ID, n>, <KEY, in>, <ID, numbers>, <SPL, :>, <KEY, print>, <SPL, (>, <String, "{n} is a number">, <SPL,)> ***Total – 15 Tokens*** *(2marks)*
3.	Given the Expression grammar of a recent high-level programming language, **E -> E * F | F + E | F** **F -> F - F | id** Find the precedence of operators with respect to the Grammar.
	*Precedence as per the Grammar: - (high), + and * both are having equal precedence (lower than -).* *(2marks)*
4.	Consider the grammar **E → E + n | E * n | n**. For a sentence **n + n * n**, identify the "handles" in the right-sentential form of the reduction technique?
	*Handles include: "n", "E+n", "E*n"* *(2marks)*

SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**

ANSWER KEY

5.	Build the derivation tree for the string **((a, a), a)** based on the following Context Free grammar **S → (L) | a** **L → L , S | S**  *(2marks)*
6.	Illustrate the process of eliminating the Left Recursion from the given grammar: **A → ABb | Bb |a** **B → Bc | Acc |b**
	A → BbA' |aA' *A' →BbA' |∈* *B →aA'ccB' | bB'* *B' →cB' | bA'ccB' | ∈* *(2marks)*
7.	Write the **LEX specification** to recognize the string w described as: ▪ w begins and ends with digit. |w| between 8 and 12 characters. ▪ w contains atleast one Upper case letter and a special character (either # or @) in the middle.
	([0-9](.[a-z].*)(.*[A-Z].*) (.*[#@].*)[0-9]).{8,12}* *(2marks)*
8.	Augment a Context Free Grammar to represent the syntax of **while & do while statements in C++.**
	S → while(E){S} | do{S}while(E); E →E *R* E | id | num *R* → < | <= | > | >= |== | != *(2marks)*
9.	Compare the behaviour of the Compiler and Interpreter upon detecting any error in the Source program during the translation.
	The compiler usually reports all syntax and semantic errors it detects in a single pass or set of passes through the source code. The program must be error-free to be successfully compiled whereas the interpreter reports errors as it encounters them during execution. This means that the execution stops at the point where the error occurs, and subsequent code is not executed. (2marks)

SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**
ANSWER KEY

10.	Consider the language of all strings over {a, b, c} containing the substring "**abcabb**". Write a regular expression that describing this language.
	*Regular Expression describing the Language: r = (a|b|c)*abcabb (a|b|c)** *(2marks)*

PART – B

ANSWER ALL THE QUESTIONS **3 X 10 = 30 Marks**

11.	Construct a LL(1) Parsing Table for the given Context Free Grammar. **A → (B) | a** **B → B , A | A**
	Steps & Split up of Marks 1. *Elimination of Left Factor* - *1 Mark* 2. *Computation of FIRST* - *2 Marks* 3. *Computation of FOLLOW* - *3 Marks* 4. *Construction of LL parsing table* - *4 Marks* ***Total*** - ***10 Marks***
12.	Design a Lexical Analyser Generator to recognize the patterns in the following LEX specification. **%%** **aba** **{action sequence A_1 for pattern P_1}** **ab*b** **{action sequence A_2 for pattern P_2}** **b+a+** **{action sequence A_3 for pattern P_3}** **%%** int main ( ) { yylex(); return 0; } int yywrap() { return 1;}
	Steps & marks Split up 1. *Construct NFA for each regular expr pattern* - *2 Marks* 2. *Combine all 4 NFAs into single NFA* - *1 Marks* 3. *NFA to DFA* - *5 marks* 4. *DFA transition diagram & table* - *2 marks* ***Total*** - ***10 marks***

SASTRA DEEMED TO BE UNIVERSITY (U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing
First CIA Exam – Aug 2024
Course Code: CSE320
Course Name: Compiler Design
Duration: 90 minutes **Max Marks: 50**
ANSWER KEY

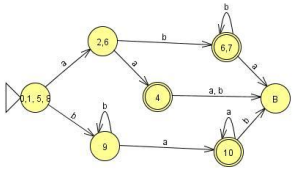
13.	Show how the Deterministic Finite Automaton is converted directly from an augmented regular expression **r=(0|1)*(0|1)**
	Steps & marks Split up 1. *Construction of Syntax tree for r* - *1 Marks* 2. *Computation of Firstpos, Lastpos for each node* - *2 Marks* 3. *Computation of Followpos for star and Cat nodes* - *3 marks* 4. *DFA States & transition* - *3 marks* 5. *Minimized DFA (transition diagram & table)* - *1 mark* ***Total*** - ***10 marks***

SOLUTION FOR PART – B QUESTIONS

11. LL(1) Parsing Table for **A → (B) | a; B → B , A | A**

N \ T	(	)	,	a	$
A	**A → (B)**	Error	Error	**A →a**	Error
B	**B → AB'**	Error	Error	**B → AB'**	Error
B'	Error	**B' →ε**	**B' →,AB'**	Error	Error

12. Lexical Analyser Generator for the Regular Expression patters in LEX code.



13. Optimized DFA for Pattern matchers: r=(0|1)*(0|1)

Pres State	Next State for 0	Next State for 1
A: {1,2,3,4}	B: {1, 2, 3, 4, 5}	B: {1, 2, 3, 4, 5}
B: {1, 2, 3, 4, 5}	B: {1, 2, 3, 4, 5}	B: {1, 2, 3, 4, 5}

Minimized DFA for r=(0|1)*(0|1) has only 2 states A & B.

***** END *****