SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

CONTINUOUS ASSESSMENT TEST - I

Program	**: B.E/B.Tech-CSE/IT**	**Max. Marks: 30**	
Course	**: Compiler Design**	**Time**	**: 1 Hour**
Course code:	**SCS1303**	**Sem**	**: V**
Batch	**: 2018-2022**	**Date**	**: 23-09-2020**

Part-A **Answer ALL the questions** **(5×2=10)**

Q.No	**Questions**	**CO (L)**
1.	Construct NFA for the regular expression: i. ab* |a* ii. a(b*|a*)	**1(5)**
2.	Compare Ambiguous and Unambiguous grammar.	**2(4)**
3.	Demonstrate the use of input buffer in Lexical Phase of Compiler.	**1(3)**
4.	Discuss the rules for eliminating Left recursion and Left Factoring with suitable example.	**2(2)**
5.	Explain the role of symbol table and error handler in a compiler.	**1(2)**

Part-B **Answer ALL the questions** **(2×10=20)**

Q.No	**Questions**	**CO (L)**
6.	Construct the Minimized DFA for **(a*|b*)*.**	**1(5)**
	(OR)	
7.	Construct the Minimized DFA for **ab(a | b)*.**	**1(5)**
8.	Consider the following Grammar: **E→ E+ T | T** **T→ T * F | F** **F→ (E) | id** a. Eliminate Left recursion. (2 marks) b. Compute FIRST() and FOLLOW(). (3 marks) c. Construct the Predictive parsing table. (2 marks) d. Parse the input string: **(id+id)*id$** (3 marks)	**2(4)**
	(OR)	
9.	Consider the following Grammar: **S→ iCtSeS | iCtS | a** **C→ b** a. Eliminate Left Factoring. (2 marks) b. Compute FIRST() and FOLLOW(). (3 marks) c. Construct the Predictive parsing table. (3 marks) d. Can an ambiguous grammar be LL(1) ?Justify. (2 marks)	**2(4)**