

**III B. Tech I Semester Supplementary Examinations, April/May -2025**  
**COMPIILER DESIGN**  
**(CSE-AI&ML)**

Time: 3 hours

Max. Marks: 70

Answer any **FIVE** Questions **ONE** Question from **Each unit**

All Questions Carry Equal Marks

\*\*\*\*\*

<b>UNIT-I</b>			
1.	a)	Describe the various phases of a compiler in detail, illustrating the transformations at each phase with a simple example.	[7M]
	b)	Explain the role of symbol table in compilation. How is it used in different phases of the compiler?	[7M]
		(OR)	
2.	a)	Differentiate between a compiler and an interpreter. Discuss the advantages and disadvantages of each.	[7M]
	b)	Explain the challenges in lexical analysis. How does it handle issues like lookahead and multi-character tokens?	[7M]
<b>UNIT-II</b>			
3.	a)	Define context-free grammar. Design a CFG for the language of balanced parentheses.	[7M]
	b)	Explain the significance of ambiguity in grammars. Show an example of an ambiguous grammar and discuss how to resolve the ambiguity.	[7M]
		(OR)	
4.	a)	What is recursive descent parsing? Write a recursive descent parser for the following grammar: $A \rightarrow aB \mid cD$ , $B \rightarrow b \mid \epsilon$ , $D \rightarrow d$ .	[7M]
	b)	Explain the concepts of FIRST and FOLLOW in parsing. Compute the FIRST and FOLLOW sets for the following grammar: $E \rightarrow TE'$ , $E' \rightarrow +TE' \mid \epsilon$ , $T \rightarrow FT'$ , $T' \rightarrow *FT' \mid \epsilon$ , $F \rightarrow (E) \mid id$ .	[7M]
<b>UNIT-III</b>			
5.	a)	Explain the differences between top-down and bottom-up parsing techniques. Provide examples of each.	[7M]
	b)	Describe the concept of operator precedence parsing. Explain how it resolves conflicts in parsing expressions.	[7M]
		(OR)	
6.	a)	What is syntax-directed translation (SDT)? Develop an SDT to evaluate simple arithmetic expressions.	[7M]
	b)	Explain the importance of intermediate code generation. Discuss different forms of intermediate code representations.	[7M]
<b>UNIT-IV</b>			
7.	a)	Discuss the different memory management techniques used during compilation.	[7M]
	b)	Explain the concept of control flow analysis in compilation. How is it used for optimization?	[7M]
		(OR)	
8.	a)	What is data flow analysis? Explain its role in compiler optimization.	[7M]
	b)	Describe common subexpression elimination as an optimization technique. Illustrate with an example.	[7M]

		<b>UNIT-V</b>	
9.	a)	Explain the issues related to code generation for different types of target machines.	[7M]
	b)	Discuss the role of register allocation in code generation. Explain different strategies for register allocation.	[7M]
		(OR)	
10.	a)	What are the different forms of object code? Compare and contrast them.	[7M]
	b)	Explain the concept of instruction selection in code generation. Give examples of how different instructions might be chosen.	[7M]

\*\*\*\*\*