

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

Regular & Supplementary Semester Examination - Summer 2024

Course: B. Tech.

Branch: Computer Engineering

Semester: VI

Subject Code & Name: BTCOC601_Y23, Compiler Design

Max Marks: 60

Date: 13 / 06 / 2024

Duration: 3 Hrs.

Instructions to the Students:

1. All the questions are compulsory.
2. The level of question/expected answer as per OBE or the Course Outcome (CO) on which the question is based is mentioned in () in front of the question.
3. Use of non-programmable scientific calculators is allowed.
4. Assume suitable data wherever necessary and mention it clearly.

	(Level/CO)	Marks
Q. 1 Solve Any Two of the following.		
A) Explain the stages involved in analyzing a source program. What are the key activities performed during the lexical, syntactic, and semantic analysis stages?	Understand Remember	6
B) What is code optimization? Describe its role in the compilation process and provide examples of common optimization techniques.	Remember Understand	6
C) Describe the middle-end of a compiler. What is its function, and why is it important to separate optimization from other phases?	Application Remember	6
Q.2 Solve Any Two of the following.		
A) Explain how regular expressions are used to specify tokens. Give an examples of regular expressions for different types of tokens, including keywords, identifiers, numbers, and string literals	Remember Understand	6
B) Describe the typical structure of a lexical specification using a language like Lex. What are the main sections of such a specification?	Remember Understand	6
C) Explain the workflow of a lexical analyzer generator from input specification to the generation of the lexical analyzer code. Provide a step-by-step description of the process.	Understand Apply	
Q. 3 Solve Any Two of the following.		
A) Write a context-free grammar for simple arithmetic expressions involving addition, subtraction, multiplication, and division. Ensure your grammar is unambiguous.	Understand Apply	6
B) What is bottom-up parsing? Describe the main techniques used in bottom-up parsing, including shift-reduce parsing.	Knowledge Understand	6
C) Construct an operator-precedence table for a given set of operators and their precedence rules. Demonstrate how this table is used to parse an arithmetic expression.	Knowledge Apply	6
Q.4 Solve Any Two of the following.		
A) Explain how SDT can be used to generate intermediate code for a simple programming language.	Understand Application	6
B) Explain the concept of top-down parsing. How does it differ from bottom-up parsing?	Remember Understand	6
C) Given the following grammar for arithmetic expressions: $E \rightarrow E + T, E \rightarrow T, T \rightarrow T * F, T \rightarrow F, F \rightarrow (E), F \rightarrow id$	Knowledge Apply	6

Write a syntax-directed translation scheme that generates three-address code for this grammar. Explain how the translation scheme works for each production rule.

Q. 5 Solve Any Two of the following.

- | | | |
|---|------------------------|----------|
| A) Explain the problem of instruction selection in code generation. How does the choice of target machine instructions impact the performance of the generated code? | Analyze
Understand | 6 |
| B) What is a basic block in the context of a compiler? Describe the process of constructing basic blocks from a given intermediate code | Remember
Understand | 6 |
| C) Describe the algorithm for calculating next-use information for a sequence of intermediate code. | Understand
Apply | 6 |

*** End ***

10-7-255.195 2024-06-13 03:36:33 UTC