Roll No. | | | | | | | | | | | | **Total No. of Pages : 02**

Total No. of Questions : 09

B.Tech. CSE (Internet of Things and Cyber Security including Block Chain Technology) (Sem.–7)

COMPILER DESIGN

Subject Code : BTCS601-18

M.Code : 93970

Date of Examination: 04-12-2023

Time : 3 Hrs. **Max. Marks : 60**

INSTRUCTIONS TO CANDIDATES :

1. **SECTION-A is COMPULSORY consisting of TEN questions carrying TWO marks each.**
2. **SECTION-B contains FIVE questions carrying FIVE marks each and students have to attempt any FOUR questions.**
3. **SECTION-C contains THREE questions carrying TEN marks each and students have to attempt any TWO questions.**

SECTION-A

1. Write briefly :

a) Define the role of the Lexical Analyzer in a compiler.

b) What is the significance of context-free grammars in syntax analysis? How are they used in parsing?

c) Explain the runtime environment of a program, including stack allocation pace and access to non-local data. Why is it important in code generation?

d) Name and briefly describe two principal sources of code optimization in compiler design.

e) Briefly describe the process of generating intermediate code. Mention the types of intermediate languages used in this phase.

f) What is the purpose of a Finite Automata in lexical analysis? How are regular expressions used to construct Finite Automata?

g) Briefly describe the LR(0) item construction process for building SLR parsing tables.

h) Discuss the concept of syntax-directed definitions and how they are used for code generation?

i) Briefly explain the Peep-hole optimization technique in code optimization.

j) In the context of syntax analysis, compare and contrast top-down parsing and bottom-up parsing techniques.

SECTION-B

2. Discuss the structure of a compiler, highlighting its key components and their functions.

3. Take a given grammar and demonstrate how to construct an LL(1) parsing table for it. Include step-by-step details.

4. Explain the concept of regular expressions and their use in specifying tokens for lexical analysis. Provide an example.

5. Describe the process of minimizing deterministic finite automata (DFA) in lexical analysis. Provide an example.

6. In the context of code generation, discuss how a compiler manages storage organization and access to non-local data on the stack.

SECTION-C

7. Discuss the process of code optimization in compilers. Explain the concept of global data flow analysis and its role in optimization. Provide an example.

8. Explain the role of YACC in generating parsers. Discuss the error handling and recovery techniques in syntax analysis.

9. In the - context of syntax-directed definitions, evaluate orders for syntax-directed definitions and discuss their role in generating intermediate code. Provide a practical example.

NOTE : Disclosure of Identity by writing Mobile No. or Making of passing request on any page of Answer Sheet will lead to UMC against the Student.