R18

Code No: 156AH

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD

B. Tech III Year II Semester Examinations, January/February - 2025

COMPILER DESIGN

(Common to CSE, ECM, CSE(DS))

Time: 3 Hours **Max. Marks: 75**

Note: i) Question paper consists of Part A, Part B.
ii) Part A is compulsory, which carries 25 marks. In Part A, Answer all questions.
iii) In Part B, Answer any one question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART – A

(25 Marks)

1.a)	What are the main components of a compiler?	[2]
b)	Define finite automata and its role in lexical analysis.	[3]
c)	Define context-free grammar with an example.	[2]
d)	Differentiate between top-down and bottom-up parsing.	[3]
e)	What are syntax-directed definitions?	[2]
f)	Define three-address code and provide an example.	[3]
g)	What is the importance of stack allocation in storage organization?	[2]
h)	Define garbage collection and its significance in run-time environments.	[3]
i)	What are the principal sources of optimization in compilers?	[2]
j)	Write the significance of constant propagation optimization.	[3]

PART – B

(50 Marks)

2.a) Discuss the structure of a compiler and its major phases.
b) Explain the process of recognizing tokens using finite automata. [5+5]

OR

3.a) Illustrate the role of regular expressions in lexical analysis with examples.
b) Discuss Optimization of DFA-Based Pattern Matchers. [5+5]

4. Consider the following grammar-
$S \rightarrow S + S$
$S \rightarrow S * S$
$S \rightarrow$ id and
Parse the string "id + id + id" using SR Parser. [10]

OR

5.a) Explain the role of syntax analysis in the compilation process with example.
b) Write the steps to construct the predictive parser table and explain with an example. [5+5]

6.a) Construct the syntax directed definition to convert infix notation into postfix notation.
b) Describe different ways of implementing intermediate code generation of a three address statement. [5+5]

OR

7.a) Explain the various evaluation orders for syntax-directed definitions.
b) Explain the significance of intermediate-code generation for control flow statements. [5+5]

8.a) Discuss the storage organization in run-time environments, including heap management.
b) Explain the methods used for accessing nonlocal data on the stack. [5+5]

OR

9.a) Write a detailed note on trace-based collection techniques.
b) Discuss the design challenges and key considerations for a code generator. [5+5]

10.a) Discuss the principal sources of optimization in machine-independent compilation.
b) Write a detailed note on the significance of data-flow analysis in optimization. [5+5]

OR

11.a) Illustrate constant propagation with examples and discuss its applications.
b) Compare and contrast machine-dependent and machine-independent optimizations. [5+5]

---ooOoo---