

Reg. No.

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Question Paper Code : 51391

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2016

Sixth Semester

Computer Science and Engineering

CS 2352/CS 62 / 10144 CS 602 – PRINCIPLES OF COMPILER DESIGN

(Regulations 2008/2010)

(Common to PTCS 2352 – Principles of Compiler Design for B.E. (Part-Time) Fifth Semester – Computer Science and Engineering – Regulations 2009)

Time : Three Hours

Maximum : 100 Marks

Answer ALL questions.

PART – A (10 × 2 = 20 Marks)

1. Define Compiler.
2. Mention the issues in a lexical analyzer.
3. What is finite automata ?
4. What is an ambiguous grammar ? Give an example.
5. What are the methods of representing a syntax tree ?
6. What are the applications of DAG ?
7. Write about dead code elimination.
8. What is peephole optimization ?
9. What is a cross-compiler ? Give an example.
10. What is constant folding ?

PART – B (5 × 16 = 80 Marks)

11. (a) (i) Explain the various phases of a compiler in detail. Also write down the output for the following expression after each phase $a := b * c - d$. (8)
- (ii) Write in detail about the cousins of the compiler. (8)

OR

- (b) (i) Explain the functions of the Lexical Analyzer with its implementation. (8)
- (ii) Elaborate the specification of tokens. (8)
12. (a) Construct Predictive Parser table for the following grammar :

$S \rightarrow (L)/a$

$L \rightarrow L, S/S$ and

Construct the behaviour of the parser on sentence (a, a) using the grammar :

$S \rightarrow (L) / a$

$L \rightarrow L, S/S$

OR

- (b) (i) Consider the grammar given below.

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id$

Construct an LR Parsing table for the above grammar. Give the moves of LR parser on $id*id+id$.

- (ii) Write note on shift reduce parsing.

13. (a) How would you generate intermediate code for the flow of control statements ? Explain with examples.

OR

- (b) (i) How Back patching can be used to generate code for Boolean expressions and flow of control statements. (8)
- (ii) Explain how the types and relative addresses of declared names are computed and how scope information is dealt with. (8)

14. (a) (i) Explain the issues in design of code generator.
- (ii) Explain DAG representation of the basic blocks with an example.

OR

- (b) Write detailed notes on Basic blocks and flow graphs.

15. (a) (i) Discuss about the following :
- (A) Copy Propagation
- (B) Dead-code Elimination and
- (C) Code motion
- (ii) Describe in detail about the stack allocation in memory management.

OR

- (b) (i) Describe in detail the source language issues.
- (ii) Write detailed notes on parameter passing.