

Reg. No. :

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

Question Paper Code : 73389

B.E./B.Tech. DEGREE EXAMINATION, APRIL/MAY 2017.

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. How will you group the phases of compiler?
2. Write the regular expression for identifier and whitespace.
3. Define an ambiguous grammar.
4. What is dangling reference?
5. What are the various ways of representing intermediate languages?
6. What is back patching?
7. Define the dead code elimination.
8. What are the issues in the design of code generator?
9. Define live variable.
10. What is data flow analysis?

PART B — (5 × 16 = 80 marks)

11. (a) (i) Write in detail about the cousins of the compiler. (8)
(ii) Explain in detail about the role of Lexical analyzer with the possible error recovery actions. (8)

Or

- (b) (i) What are the phases of the compiler? Explain the phases in detail. Write down the output of each phase for the expression $a := b + c * 50$. (10)
- (ii) Elaborate specification of tokens. (6)
12. (a) (i) Distinguish between context free grammars and regular expressions. (8)
- (ii) What are the conflicts possible during shift-reduce parsing? Explain it with example. (8)

Or

- (b) (i) What are the storage allocation strategies? Explain them with example. (8)
- (ii) Distinguish between static and dynamic storage allocations. (8)
13. (a) (i) Distinguish between quadruples and triples with example. (8)
- (ii) What are the rules for type checking? Give an example. (8)

Or

- (b) (i) State and explain the algorithm for unification. (8)
- (ii) Explain the one pass code generation using back patching with example. (8)
14. (a) (i) Explain in detail about the various issues in design of code generator. (10)
- (ii) Write an algorithm to partition a sequence of three address statements into basic blocks. (6)

Or

- (b) (i) Explain the code-generation algorithm in detail. (8)
- (ii) Construct the DAG for the following basic block. (8)

$d := b * c$

$e := a + b$

$b := b * c$

$a := e - d$

15. (a) Discuss in detail about global data flow analysis. (16)

Or

- (b) (i) Explain the three techniques for loop optimization with examples. (12)
- (ii) Explain constant folding and copy propagation with examples. (4)