LIB
7/1/16 AN

Reg. No. : | | | | | | | | | | | | |

**Question Paper Code : 21391**

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2015.

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. State the interactions between the lexical analyzer and the parser.

2. What are the components of Lex?

3. Define the error recovery strategies?

4. State the kinds of data that appear in activation record?

5. What are synthesized attributes?

6. What is a short circuit code?

7. Define the dead code elimination.

8. What are the issues in the design of code generator?

9. What are the global common sub expressions?

10. What are the control-flow constraints?

PART B — (5 × 16 = 80 marks)

11. (a) (i) State and explain the architecture of a transition–diagram–based lexical analyzer. (8)

(ii) How to minimize the number of states of DFA. Explain it with example. (8)

Or

(b) (i) Explain the procedure for construction of an NFA from a regular expression. (8)

(ii) What are the functions computed from the syntax tree. Explain each function with example. (8)

12. (a) (i) Distinguish between context free grammars and regular expressions. (8)

(ii) What are the conflicts 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) Generate code for the following three-address statements assuming stack allocation where register SP points to the top of the stack. (10)

```
call p
call q
return
call r
return
return
```

(ii) What are local common sub expressions? Explain the procedure to eliminate the local common sub expressions. (6)

Or

(b) (i) Explain the procedure for register allocation by graph coloring with an example. (8)

(ii) Explain the steps carried out for generating code from dags with suitable example. (8)

15. (a) (i) Explain the procedure to find the induction variable in loops and optimize their computation. What is the function of strength reduction? (8)

(ii) What is peephole optimization? State and explain the characteristic of peephole optimization. (8)

Or

(b) (i) What is data flow abstraction? Explain it with a program illustrating the data flow abstraction. (8)

(ii) What is live-variable analysis? Explain it with example. (8)

---