

[5254]-662

B.E. (Computer Engineering)
PRINCIPLES OF MODERN COMPILER DESIGN
(2012 Pattern) (Semester - I)

*Time : 3 Hours]**[Max. Marks : 70**Instructions to the candidates:*

- 1) *Answer Questions. 1 or Q. 2, Q. 3 or Q. 4, Q. 5 or Q. 6, Q. 7 or Q. 8 Q. 9 or Q. 10*
- 2) *Neat diagrams must be drawn wherever necessary.*
- 3) *Assume suitable data, if necessary.*
- 4) *Figures to the right indicate full marks.*

- Q1)** a) Why compilation phases are divided into front-end and back-end? What are the advantages? **[4]**
- b) Give syntax directed definition for any example arithmetic expression. **[6]**

OR

- Q2)** a) What is YACC? Give format of Yacc specification file. **[4]**
- b) Write the syntax directed translation scheme for generating Intermediate code for array assignment statement. **[6]**

- Q3)** a) Explain the terms phase and pass related to compiler. **[2]**
- b) Construct LL (1) parsing table for the following grammar. **[8]**

Terminals = {id, num, while, print, >, {, }, ;, (,)}

Nonterminal = {S, E, B, L}

- Rule =
- i) $S \rightarrow \text{print } (E)$
 - ii) $S \rightarrow \text{while } (B)S$
 - iii) $S \rightarrow \{L\}$
 - iv) $E \rightarrow \text{id}$
 - v) $E \rightarrow \text{num}$
 - vi) $B \rightarrow E > E$
 - vii) $L \rightarrow SL$
 - viii) $L \rightarrow \epsilon$

Start Symbol = S

P.T.O.

OR

Q4) a) Enlist the operations performed on symbol table. [2]

b) Construct SLR (1) parsing table for the following grammar. [8]

$S \rightarrow aAb|bB$

$A \rightarrow Aa| \epsilon$

$B \rightarrow Bb| \epsilon$

Q5) a) What do you mean by common sub-expression? Discuss the algorithm for elimination of common sub-expression. [6]

b) Discuss peephole optimization techniques. [6]

c) What is DAG? with suitable illustrations explain the role of DAG in code generation phase. [6]

OR

Q6) a) Discuss following optimizations with example. [6]

i) Strength reduction

ii) Dead code elimination.

b) What do you meant by 'Next Use' information? How it is computed?[6]

c) Explain the algorithm for generating code from labeled tree. [6]

Q7) a) Explain the different translation schemes to remove syntactic sugar from Haskell program. [6]

b) Explain following features of object oriented languages related to compiler design. [6]

i) Overloading

ii) Inheritance.

c) Discuss features of Java CC compiler. [4]

OR

- Q8)** a) Discuss following with respect to object oriented languages. [6]
i) Type checking.
ii) Type coercion.
- b) Explain following with respect to functional languages. [6]
i) Referential transparency.
ii) Lazy evaluation.
- c) What is activation record? Explain possible structure of an activation record? [4]

- Q9)** a) Discuss parallel programming models. [6]
- b) Write short notes: [6]
i) g++
ii) NVCC
iii) LLVM
- c) Compare processes and threads. [4]

OR

- Q10)** a) Discuss issues in message passing techniques. [6]
- b) Explain following concepts related to automatic parallelization. [6]
i) Data dependencies.
ii) Loop transformations.
- c) What is Interpreter? Explain Dalvik. [4]

