[5354]-672
B.E. (Computer Engineering)
PRINCIPLES OF MODERN COMPILER DESIGN
(2012 Pattern)

Time : 2½ Hours]

[Max. Marks : 70

Instructions to the candidates:
1) *Neat diagrams must be drawn wherever necessary.*
2) *Figures to the right indicate full marks.*
3) *Assume suitable data, if necessary.*

Q1) a) Discuss the action taken by every phase of compiler on following string:**[6]**

A= B * (C+D)–E

b) Translate the following statements into three address code: **[4]**

x = A [i] * m/n^p+ g & q [i]

OR

Q2) a) What is Lex? How tokens are generated in Lex? **[6]**

b) Explain: Error detection and recovery in YACC? **[4]**

Q3) a) Write Syntax Directed Translation scheme for boolean expression? **[4]**

b) Compare: Quadruple, Triple, Indirect Triple. **[6]**

OR

Q4) a) Differentiate between SLR, LR(K) and LALR parser. **[6]**

b) Explain advantages of dividing three address statements into basic blocks. **[4]**

Q5) a) Explain: Issues in code generation. **[6]**

b) Write the need for code optimization. **[4]**

c) Explain labelling algorithm for tree. **[8]**

P.T.O.

OR

Q6) a) Discuss the principle sources of code optimization, give proper examples wherever necessary. **[8]**

b) What are different terms related to simple code generation algorithm.**[4]**

   i) Register descriptor.

   ii) Address descriptor.

c) What is next use information? Explain its use in code generation. **[6]**

Q7) a) Explain type checking with respect to context handling. **[6]**

b) What is meant by desugaring? Why is importance of it. **[6]**

c) Compare between Structure and Union type. **[4]**

OR

Q8) a) Write note on Java CC. **[6]**

b) Write note on Desugaring. **[6]**

c) Compare register oriented and stack oriented architectures. **[4]**

Q9) a) Explain Dynamic compilation. **[6]**

b) Write note on data parallelism. **[6]**

c) Explain marshalling and unmarshalling of messages. **[4]**

OR

Q10) a) Explain cross compilation using XMLVM. **[6]**

b) Write note on nVidia cuda compiler. **[6]**

c) Write difference between GCC and G++. **[4]**

○○○