Total No. of Questions : 10]

SEAT No. : □

**P2010**

[Total No. of Pages : 3

## [4859]-1071

## B. E. (Computer Engineering) (End sem.) (Semester - I)

# PRINCIPLES OF MODERN COMPILER DESIGN

## (2012 Pattern)

*Time : 3 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) Why the analysis phase of compiler is normally separated into lexical analysis and parsing phases? **[4]**

b) Explain the following in brief: **[6]**

i) Operating system support for compiler.

ii) Memory management.

OR

*Q2)* a) Explain frontend phases of compiler with example. **[6]**

b) Describe role of Regular expression in lexical analysis. What is lookahead operator? **[4]**

*Q3)* a) Compute First and Follow sets for following grammar. Also generate predictive parsing table for this grammar. **[8]**

$S \rightarrow ABa \mid bCA$

$A \rightarrow cBCD \mid \varepsilon$

$B \rightarrow CdA \mid ad$

$C \rightarrow eC \mid \varepsilon$

$D \rightarrow bSf \mid a$

b) Enlist different error recovery methods in parsing. **[2]**

OR

*P.T.O.*

*Q4)* a) Construct SLR parsing table for following grammar. **[7]**

$S \rightarrow xAy \mid xBy \mid xAz$

$A \rightarrow aS \mid q$

$B \rightarrow q$

Also show moves by parser for the string "xqy"

b) Explain how syntax directed translation scheme is implemented with topdown parser. **[3]**

*Q5)* a) What is code optimization? Differentiate among local, global and loop optimization. **[6]**

b) Generate DAG from following 3 address statements. **[5]**

T1 = b + c

T2 = d * e

T3 = d + c

T4 = T2 * T3

T5 = T4 * f

X = T1 – T5

Traverse this DAG using topological order and list the optimized code. Which optimization is carried out in this process?

c) What is Register Allocation and Assignment problem? **[4]**

d) Why redundant load and store are undesirable in code generation process? **[3]**

OR

*Q6)* a) Explain following data flow properties: **[6]**

Available expression Reaching definitions live variables.

b) Discuss following optimizations with examples:

Variable propagation, code movement. **[4]**

c) Which are various issues related to code generation. **[4]**

d) Write short note: Peephole optimization. **[4]**

*Q7)* a) Explain the structure of a functional compiler. Discuss various issues related to compilation of functional languages by taking example of Haskell. **[8]**

b) Explain following with respect to Object Oriented languages: **[6]**

Type checking, Type equivalence, Type coercion.

c) What is JavaCC? **[2]**

OR

*Q8)* a) Explain following points with respect to functional language Haskell. Offside rule, Pattern matching, Referential transparency, Polymorphic typing. **[8]**

b) Consider following features of Object Oriented languages:

Overloading and Polymorphism. Explain what changes are needed in compiler design to accommodate these features. **[6]**

c) What is lazy evaluation in functional languages? **[2]**

*Q9)* a) What is interpreter? Explain JVM as an example of interpreter. **[4]**

b) Discuss issues related to parallel compiler. Explain with respect to NVCC. **[6]**

c) Explain following points for parallel object oriented languages:

Object location, object migration, object replication. **[6]**

OR

*Q10)* Write short notes: **[16]**

i) Tuple space

ii) XML VM

iii) Processes and threads

iv) JIT

⊙⊙⊙