Code :R7320502

$\boxed{1}$

III B.Tech II Semester(R07) Regular & Supplementary Examinations, April/May 2011
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours
Max Marks: 80

Answer any FIVE questions
All questions carry equal marks
⋆ ⋆ ⋆ ⋆ ⋆

1. (a) What is language translator and functions of language translators?
   (b) How does the lexical analyzer identifies the tokens.

2. (a) Discuss in detail about different orders of derivation in parsing.
   (b) What is the role of Parser? Explain in detail.

3. (a) Write algorithm for the operator precedence parsing.
   (b) Discuss in detail about LR parser.

4. (a) Explain about syntax directed definitions in detail.
   (b) Explain in detail about the construction of syntax trees.

5. Explain the storage allocation in unstructured languages.

6. Explain the principle sources of optimization with suitable examples.

7. Write about Data flow analysis of structural programs.

8. (a) What is a basic block? With suitable example - discuss various transformations on the basic block.
   (b) Explain the concept of Register allocation and assignment.

⋆ ⋆ ⋆ ⋆ ⋆

Code :R7320502

| 2 |

III B.Tech II Semester(R07) Regular & Supplementary Examinations, April/May 2011
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours   Max Marks: 80

Answer any FIVE questions
All questions carry equal marks
★★★★★

1. (a) Explain format for the input or source file of LEX.

    (b) Discuss in detail about lexical errors.

2. (a) Compare regular expressions Vs Context free grammar.

    (b) Discuss in detail about top down parsing.

3. (a) How to identify the operator precedence relations from associativity and Precedence.

    (b) Explain about the construction of precedence functions.

4. (a) Give syntax directed definition of a simple desk calculator.

    (b) Discuss in detail about different intermediate forms of source program.

5. Explain in detail the implementation of a simple stack - allocation scheme.

6. Write an algorithm to construct DAG. Explain it with example.

7. (a) What are basic blocks and flow graphs? Explain with example.

    (b) Define induction variable. Give examples. Explain its use.

8. (a) Explain the translation of assignment statements.

    (b) What do you mean postfix translations?

★★★★★

Code :R7320502

III B.Tech II Semester(R07) Regular & Supplementary Examinations, April/May 2011
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours
Max Marks: 80

Answer any FIVE questions
All questions carry equal marks
⋆ ⋆ ⋆ ⋆ ⋆

1. (a) What is a LEX tool? How LEX programs are used to perform lexical Analysis.
   (b) Explain procedure for obtaining the regular expression from finite automata.

2. (a) Explain recursive descent parsing.
   (b) Write an algorithm for construction of predictive parsing table.

3. (a) What is the significance of LR parsing and what are its advantages?
   (b) Discuss in detail about error recovery in Parsing.

4. Give a translation scheme for checking that the same identifier doesn't appear twice in the list of identifiers

5. Explain the linear list hash table and symbol table mechanism with examples.

6. Explain in detail about the DAG representation of Basic Blocks.

7. (a) What are dominators? Explain with example.
   (b) Write the algorithm for Code motion.

8. Discuss the problems in code generation.

⋆ ⋆ ⋆ ⋆ ⋆

Code :R7320502

$\boxed{4}$

III B.Tech II Semester(R07) Regular & Supplementary Examinations, April/May 2011
COMPILER DESIGN
(Computer Science & Engineering)

Time: 3 hours

Max Marks: 80

Answer any FIVE questions
All questions carry equal marks
⋆ ⋆ ⋆ ⋆ ⋆

1. (a) Write about lexical analyzer generator.

   (b) Give regular definitions for unsigned numbers in PASCAL.

2. (a) Explain about error recovery in predictive parsing.

   (b) Explain in detail about the elimination of ambiguity in grammar.

3. (a) Discuss procedures closure and goto for constructing sets of LR(1) items

   (b) Implement stack of SLR parser for the input string id* id +id

4. Discuss in detail about synthesized and inherited attributes.

5. How would you map names to values? Explain with example.

6. Explain in brief about intermediate code optimization algorithms.

7. List the criteria for selecting a code optimization technique and explain any three Optimization techniques.

8. Describe in detail about a simple code generator with the appropriate algorithm.

⋆ ⋆ ⋆ ⋆ ⋆