USN [   ]   **10IS662**

Sixth Semester B.E. Degree Examination, Dec.2014/Jan.2015

Compiler Design

Time: 3 hrs.   Max. Marks:100

Note: *Answer FIVE full questions, selecting*
at least TWO questions from each part.

PART – A

1 a. Describe the various phases of a compiler. Write down the translation for an assignment statement like: position = initial + rate * 60. Indicate the outputs of each phase. **(12 Marks)**

   b. Construct the transition diagrams for a set of keywords like begin, end, if, then and else, and identifiers and constants along with a minimum set of relational operators. **(08 Marks)**

2 a. How left recursion can be eliminated from grammars? Write down the simple arithmetic expression grammar and rewrite the grammar after removing left recursion. **(05 Marks)**

   b. What is left factoring? Rewrite the following grammar after "left factored":

      S → iEtS | iEtSeS | a

      E → b   **(05 Marks)**

   c. Broadly classify parsers. How top-down parsing and bottom-up parsing be subdivided? Briefly explain the merits of each. **(05 Marks)**

   d. Briefly explain the implementation of a table driven predictive parser. **(05 Marks)**

3 a. Define LL(1) grammars. Test whether the following grammar is LL(1) or not, and construct a predictive parsing table for it.

      S → AaAb | BbBa

      A → ∈

      B → ∈   **(10 Marks)**

   b. Construct the LR(0) item set for the grammar:

      E → E + T / T

      T → T * F / F

      F → (E) / d

      Also draw the DFA for the et of LR(0) items found. **(10 Marks)**

4 a. Construct the canonical LR(1) items and the GOTO graph as well as the canonical LR(1) parsing table for the following augumented grammar:

      S′ → s

      S → CC

      C → cC | d   **(16 Marks)**

   b. For the table in question 4(a), construct the LALR parsing table. **(04 Marks)**

PART – B

5 a. Explain the concept of syntax-directed definitions. What is synthesized attributes and inherited attributes? **(06 Marks)**

   b. Construct the syntax tree, parse tree, and the annotated parse tree for the input string say 5 * 6 + 7; using suitable context-free grammar. **(06 Marks)**

   c. Construct a dependency graph for the declaration float id1, id2, id3. **(08 Marks)**

6 a. Draw the syntax tree and DAG for the expression $(a * b) + (c - d) * (a * b) + b$. **(08 Marks)**
    b. Represent the following assignment namely $a = b * - c + b * - c$; in its syntax tree form, three-address code, quadruples and triples representation. **(12 Marks)**

7 a. Discuss the general structure of activation record. **(08 Marks)**
    b. What is meant by calling sequence and return sequence? List the calling sequence design principles. **(08 Marks)**
    c. Write a note on garbage collection. **(04 Marks)**

8 a. List and briefly explain the design issues of a code generator. **(10 Marks)**
    b. With example explain common subexpression and dead code elimination methods. **(10 Marks)**

* * * * *