Reg. No. : | | | | | | | | | | |

Question Paper Code: 31862

B.E. / B.Tech. DEGREE EXAMINATION, NOV 2016

Sixth Semester

Information Technology

01UIT602 – COMPILER DESIGN

(Regulation 2013)

Duration: Three hours Maximum: 100 Marks

Answer ALL Questions

PART A - (10 x 2 = 20 Marks)

1. Interpret the two parts of a compilation and List the phases that constitute it.
2. Label the issues to be considered in the design of lexical analyzer.
3. State the role of a parser.
4. Illustrate the functions used to create the nodes of syntax tree.
5. Define back patching? List its functions for One-pass code generation.
6. State what do you mean by handle pruning.
7. List the applications of DAG.
8. Summarize the different data flow properties.
9. Recall code optimization.
10. Define Peephole optimization. Write the characteristics of peephole optimization.

PART - B (5 x 16 = 80 Marks)

11. (a) (i) Explain the needs of grouping of phases of compiler. (8)

(ii) Depict the language processing system and describe about the cousins of the compiler. (8)

Or

(b) Discuss in detail the various phases of a compiler for translating a source statement to object code. (16)

12. (a) (i) Given the following grammar *S->AS|b*, *A->SA|a*. Construct a SLR parsing table for the string *baab*. (10)

(ii) Explain the Storage organization strategies with examples. (6)

Or

(b) Construct a SLR parsing table for the following grammar and parse the string *baab*.

S->AS|b

A->SA|a. (16)

13. (a) (i) Give a short note on procedure calls. Revise the following grammar for a simple procedure call statement *S->call id* (enlist). (8)

(ii) Recall the various methods of implementing three address statements? Create the intermediate code representation for the expression a or b and not c. (8)

Or

(b) (i) Demonstrate the storage allocation strategies used in run time environments. (8)

(ii) Design a simple code generator and explain with example. (8)

14. (a) Infer the types of transformation that can be applied to basic blocks? Discuss it with neat examples. (16)

Or

(b) (i) Write detailed notes on basic blocks and flow graphs. (8)

(ii) How would you construct a DAG for a basic block? Explain with an example. (8)

15. (a) Write short notes on: (i) Peep hole optimization (ii) Issues in code generation. (16)

Or

(b) (i) Recall about principal source of optimization. (8)

(ii) Describe about global data flow analysis with examples. (8)

———————