Exam.Code:0918
Sub. Code: 6410

2014

B.E. (Computer Science and Engineering)

Sixth Semester

CS-604: Complier Design

Time allowed: 3 Hours **Max. Marks: 50**

NOTE: Attempt five questions in all, including Question No. I which is compulsory and selecting two questions from each Section.

x-x-x

I. Write short answers of the following:

a. Differentiate between phase of a compiler and pass of a compiler. What are the conditions that favor design of a multi-pass compiler?

b. List common forms of intermediate code representation.

c. What are the rules to check whether a given grammar is LL(1) or not?

d. What is loop unrolling and loop jamming? Give example.

e. What is operator grammar? **(2 marks each)**

Section-A

II.

a. What is left recursion? What are the problems that arise due to left recursion in the design of top-down parsers? Write an algorithm to remove left recursion from a given grammar. Eliminate the left recursion from the grammar:

$S -> (L)|a$

$L -> L,S|S$

b. What are translators? List various translators. How is compiler different from other translators? **(6,4)**

III.

a. Construct the LALR parsing table for the following grammar:

E→ E+T | T
T→ TF | F
F→ F* | a | b

b. Describe in brief various error recovery techniques used by parsers. **(6, 4)**

IV.

a. Construct the FIRST and FOLLOW set for the non-terminals in the following grammar:

S→Abb|C
A→aA|b
C→ab|cde

b. Define handle. What are the issues involved in handle pruning? How is handle pruning implemented in an operator precedence parser? **(5, 5)**

Section-B

V.

a. What do you mean by intermediate code? Describe in detail the advantages of generating the intermediate code.

b. What is symbol table? What is its use? Describe in brief various data structures used for storing symbol table. **(5, 5)**

VI.

a. What are the issues that must be taken care off while designing a code generator?

b. Describe in brief stack allocation strategy along with its merits and demerits. **(5, 5)**

VII.

a. Describe in detail how synthesized and inherited attributes in the semantic rules are evaluated.

b. Describe structure preserving transformations that can be applied on the basic blocks to optimize code.. **(5, 5)**

x-x-x