INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

Department of Computer Science & Engineering

B.Sc. in CSE Semester Final Examination, Spring 2019

Course Code: **CSE-4801** Course Title: **Compiler**

Full Marks: **50** Time: **2 Hours 30 minutes**

---

*[Answer any **two** questions from **Group-A** and any **three** questions from **Group-B**; Separate answer script must be used for Group-A and Group B (Figures at right margin illustrate marks)]*

Group – A [2×10 = 20 Marks]

1. a) Mention the different types of errors that occur in compiler and explain different error-recovery strategies. 3

   b) Why do we need Left Factoring? What will be the following grammar after Left Factoring: 3

   $S \rightarrow iEtS \mid iEtSeS \mid a$

   $E \rightarrow b$

   c) Find out First and Follow for all the non-terminals of the following grammar: 4

   S→ ABCD

   A→ xy | ϵ

   B→ uv

   C→ t

   D→ pq | ϵ

2. a) Prepare a predictive parsing table for the following grammar: 6

   E→ TE'

   E'→ +TE' | ϵ

   T→ FT'

   T'→ *FT' | ϵ

   F → (E) | id

   b) Using the parsing table show all the moves made on the string **id + id * id** 4

3. a) Construct a canonical collection of LR(1) items from the following grammar: 5

   S→ CC

   C → cC | d

   b) Prepare an LALR parsing table for the grammar in question 3(a). 4

   c) What is the difference between top down and bottom up parsing? 1

Group – B [3×10 = 30 Marks]

4. a) What do you mean by the inherited attributes? Give the semantic rules for the following simple type definition grammar's productions: 3

   $D \rightarrow TL$

   $T \rightarrow \mathbf{int} \mid \mathbf{float} \mid \mathbf{char} \mid \mathbf{double}$

   $L \rightarrow L_1, \mathbf{id} \mid \mathbf{id}$

   b) Give an example of type checking of expression in compiler design. 1

   c) What is dependency graph? Construct the dependency graph for the input string "**int x, y, z**" by considering the grammar of 4(a). 3

   d) Write the main basic differences of synthesized attributes and inherited attributes. Draw the annotated parse tree for the input expression **"3*5+4n"** for considering the following grammar: 3

$$E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow (E) \mid \boldsymbol{digit}$$

5. a) Translate the arithmetic expression $o := m * (-n) + m * (-n) + m * n$ into 5

i) Syntax tree
ii) Three-address code
iii) Triples
iv) DAG
v) Quadruples

b) Define activation records with general fields. 3

c) Explain dynamic and stack allocation of run-time storage strategies in a compiler design. 2

6. a) Consider the following C codes and answer the questions (i) to (iii):

```
void production(int a[],int b[]){
   int prod=0,i=1;
do{
        prod = prod + a[i] * b[i];
        i = i + 1;
   }while (i<= 20);
}
```

i) Translate the C code into three-address code. 4

ii) Identify the basic block in three-address code. 2

iii) Construct the flow graph from the three-address code. 1

b) What do you mean by peephole optimization? Generate the assembly code for the following sequence of statements: $a := b + c \quad d := a - c$

7. a) Define Directed Acyclic Graph (DAG). Write the instructions to construct Abstract Syntax Tree and construct the DAG for the expression "$a + a * (b - c) + (b - c) * d$". 4

b) What is activation tree? Draw the activation tree for the recursive factorial function that input integer n=5. 2

c) Fill in the blanks with appropriate word/words: 1

i) Generally, Types definition grammar use _____ attributes.
ii) In DAG, the total number of nodes is ____.

d) Construct the dag for the following basic block. 3

$d := b * c$
$e := a + b$
$b := b * c$
$a := e - d$

[END]