B.Sc. Engg. CSE 8th Semester

17 August 2018 (Afternoon)

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION — **SUMMER SEMESTER, 2017-2018**

DURATION: 1 Hour 30 Minutes — **FULL MARKS: 75**

CSE 4801: Compiler Design

Programmable calculators are not allowed. Do not write anything on the question paper.

There are 4 (four) questions. Answer any 3 (three) of them.

Figures in the right margin indicate marks.

---

1. a) Discuss the role of *symbol table manager* and *error handler* during compilation. 10
   b) Write down the algorithm for SLR parsing method along with parser block diagram. 10
   c) Discuss on various sentential forms. 5

2. a) In a proposed programming language variables are needed to be declared as per following format- 10

   data_type var1,var2,var3;

   Keywords to declare various types of variables in the proposed language are *integer*, *char* and *float*.

   Design a grammar to recognize multiple lines of variable declarations in prescribed format.
   b) Discuss on various recovery strategies for syntax analysis phase. 10
   c) Explain the drawbacks of recursive-descent parsing. 5

3. a) Write short notes on the following lex variables/functions. 12

   yylex(), yytext, yyleng, yywrap(), yyin, ECHO
   b) Write a lex program which will search email addresses inside a text file. File name will be given to the program as an argument. The program should print following information in new line for each of the detected email address: 13

   email_address, line_number, position_in_line

4. a) Show that the following grammar is ambiguous- 8

   stmt → if expr then stmt
   | if expr then stmt else stmt
   | other

   Rewrite the grammar by eliminating the ambiguity.
   b) Find the sets of FIRST and FOLLOW for each non-terminal of the following grammar: 7

   S → A a
   A → B D
   B → b | ε
   D → d | ε

   c) An SLR parse table is shown in Figure 1 and respective grammar is shown in Figure 2. Show the moves of a SLR parser for validation of input: **id*(id+id(** . 10

STATE	ACTION						GOTO		
	id	+	*	(	)	$	*E*	*T*	*F*
0	s5			s4			1	2	3
1		s6				acc			
2		r2	s7		r2	r2			
3		r4	r4		r4	r4			
4	s5			s4			8	2	3
5		r6	r6		r6	r6			
6	s5			s4				9	3
7	s5			s4					10
8		s6			s11				
9		r1	s7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

Figure 1

1. E → E + T
2. E → T
3. T → T * F
4. T → F
5. F → (E)
6. F → id

Figure 2