Hajee Mohammad Danesh Science and Technology University, Dinajpur
Department of Computer Science and Engineering
B. Sc. in CSE
Semester Final Examination 2017 (Jan-Jun)
Level 3 Semester I, Course Code: CSE 307, Credit: 3.0
Course Title: Compiler

Time: 3 Hours                                                                                     Total Marks: 90

*[N.B. The figure in the right margin indicates the marks allocated for respective question, all the portions of each question must be answered consecutively]*

## Section-A
Answer any <u>THREE</u>

1. a) What is a compiler, what does it do? — 2

   b) Define meta-language? Explain the use of T-diagram to explain a compiler. — 3

   c) Explain the following terms in short: Preprocessor, Assembler, Linker, and Loader. — 6

   d) Write the names of different phases of a compiler. — 4

2. a) What is Context-Free Grammar (CFG)? Describe its different components. — 1+4

   b) Consider following grammar: **E→E+E|E-E|1|2|3|4|5|6|7|8|9|0**
   Is it an ambiguous grammar? Why? If yes re-write the grammar to eliminate the ambiguity and also demonstrate the updated grammar. — 1+3 +3

   c) What is derivation of a CFG? Explain with example. — 3

3. a) Explain the use of semantic rules in compiler construction with example. — 4

   b) Write the difference(s) between bottom-up and top-down parser. — 4

   c) *"A predictive parser is a recursive descent parser that does not require backtracking"*. Explain the statement with proper example. — 4

   d) When is a grammar said to be left recursive? Explain with example. — 1+2

4. a) Describe the role of the input buffering process during lexical analysis. — 4

   b) Write the difference(s) between syntactic errors and semantic errors. — 3

   c) Explain the basic working principle of panic mode error recovery and mention its benefits. — 3+1

   d) Why elimination of left recursion from a grammar is important? Explain the process with example. — 1+3

## Section-B

Answer any THREE

1. a) Describe the role of **FIRST** and **FOLLOW** of a grammar in compiler design. 3

   b) Write the rules of computing **FIRST(A)** for all grammar symbol **A**. 5

   c) Define LL and LR parser. 2

   d) Find LL derivations of "**1*2+3**" considering the following grammar. 5

```
E→TP
P→+E| ε
T→FM
M→*T | ε
F→N
N→1|2|3
```

2. a) Write the difference(s) between abstract and concrete parse tree with example. 4

   b) Draw annotated parse tree for the expression **(3+4)*(5+6)n** considering the following CFG. Assume the symbols '+' and '*' for addition and multiplication respectively. 7

```
L→En
E→E+T|T
T→T*F|F
F→(E)|D
D→0|1|2|.....|8|9
```

   c) How operator precedence can be maintained in a CFG? Explain your answer using the CFG given in question 2(b). 4

3. a) Write syntax directed definition (SDD) for the CFG given in question 2(b). Assume the symbols '+' and '*' for addition and subtraction respectively. 5

   b) How quadruples and triples can be used to generate tree address code? Explain the procedure with proper example. 2+3

   c) Translate the arithmetic expression **a+b-(b+c)** into quadruples and triples.. 5

4. a) What is target code? When do compilers generate target code? 1+2

   b) What is flow graph? State its significance in code generation. 1+3

   c) Mention the key benefits of code optimization by a compiler. 4

   d) How do compilers usually perform "Dead Code Elimination"? Describe with proper example. 4