DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF BARISHAL

FINAL EXAMINATION-2025
Course Title: Compiler Design and Construction
Course Code: CSE-4103
4th Year, 1st Semester, Session (Admission): 2019-20

Time: 3 hours **Marks:** 60

Answer any Five Questions from the followings.

1. a) In Compiler, syntax-directed checker and translator do the static and dynamic type checking. Distinguish between static and dynamic checking with example(s). 5

   b) What will be the output of the lexical, syntax and sematic analyzers for the following expression: 5
   if(y%4==0) {printf("This may be a leap year")}

   c) What is the advantage of front end and back end model of compiler? 2

2. a) What is a *dangling else*? 2

   b) Compiler design is not only about compilers, and many people use the technology learned by studying compilers in school, yet have never, strictly speaking, written (even part of) a compiler for a major programming language. Compiler technology has other important uses as well. Additionally, compiler design impacts several other areas of computer science. Review the three most important interactions and applications of the technology. Give example for these applications. 4

   c) Let, a Context-free grammar for simple expressions: G1 = <{*list,digit*}, {+,-,0,1,2,3,4,5,6,7,8,9}, *P*, *list*> with production 6

   $P = list \rightarrow list + digit$
   $list \rightarrow list - digit$
   $list \rightarrow digit$
   $digit \rightarrow 0 \mid 1 \mid \ldots \mid 9$

   Suppose we use a single nonterminal *string* and consider the following context-free grammar: G2 = <{*string*}, {+,-,0,1,2,3,4,5,6,7,8,9}, *P*, *string*> with production *P* = *string* → *string* + *string* | *string* - *string* | 0 | 1 | ... | 9.
   Which grammar (from G1 and G2) is ambiguous because more than one parse tree represents the string 10-6+3 ?

3. a) Differentiate the terms (i) Yacc compiler, (ii) Lex compiler, and (iii) C compiler 6

   b) When a production for nonterminal *A* starts with a self-reference then a predictive parser loops forever. For example, $A \rightarrow A\alpha \mid \beta \mid \gamma$. 6
   We can eliminate left recursive productions by systematically rewriting the grammar using right recursive productions. Analyse this procedure.

4. a) Analyze the terms: *Recursive descent parsing* and *Predictive parsing* 4

   b) The following grammar, which abstracts the dangling-else problem is considered as ambiguous grammar. The grammar is ambiguous and the ambiguity is manifested by a choice in what production to use when an e (else) is seen. 8

   $S \rightarrow i\ E\ t\ S\ S_R \mid a$
   $S_R \rightarrow e\ S \mid \varepsilon$
   $E \rightarrow b$

   How to make unambiguous ?

5. a) Based on LL(1) predictive parsing table for the following grammar, describe how to do panic mode recovery and phrase level recovery. 6

$E \rightarrow T E_R$
$E_R \rightarrow + T E_R \mid \varepsilon$
$T \rightarrow F T_R$
$T_R \rightarrow * F T_R \mid \varepsilon$
$F \rightarrow (E) \mid \mathbf{id}$

$A \rightarrow \alpha$	FIRST(α)	FOLLOW(A)
$E \rightarrow T E_R$	(id	$)
$E_R \rightarrow + T E_R$	+	$)
$E_R \rightarrow \varepsilon$	ε	$)
$T \rightarrow F T_R$	(id	+ $)
$T_R \rightarrow * F T_R$	*	+ $)
$T_R \rightarrow \varepsilon$	ε	+ $)
$F \rightarrow (E)$	(	* + $)
$F \rightarrow \mathbf{id}$	id	* + $)

6

b) The following is a substring of grammar symbols:
a b b c d e
Prove that this substring shows the property of "Handle".

Grammar:
$S \rightarrow \mathbf{a}\ A\ B\ \mathbf{e}$
$A \rightarrow A\ \mathbf{b}\ \mathbf{c} \mid \mathbf{b}$
$B \rightarrow \mathbf{d}$

6. a) In *LR(k)* Parsers, Shift/Reduce decisions can be ambiguous. We can use a DFA for Shift/Reduce decisions in order to resolve the ambiguous issue. Describe it using Stack. 8

Grammar:
$S \rightarrow C$
$C \rightarrow A\ B$
$A \rightarrow \mathbf{a}$
$B \rightarrow \mathbf{a}$

b) Describe algorithm for LR Parser. 4

7. a) A good compiler should assist in identifying and locating errors. Describe about (a) Lexical errors, (b) Syntax errors, (c) Static semantic errors, (d) Dynamic semantic errors, (e) Logical errors. 6

b) Describe typical tasks of the lexical analyzer. 3

c) If the following expression is passed to the lexical analyzer, then what is the output? 3
y := 31 + 28*x

8. a) Consider the following grammar and remove left recursion. 5
$A \rightarrow ABd/Aa/a$
$B \rightarrow Be/b$

b) Discuss the position of a Code Generator in the compiler model using figure. 3

c) Let, you would like do parsing for syntax analyzer. Write down a table to show *FOLLOW(A)*. 4

$E \rightarrow T E_R$
$E_R \rightarrow + T E_R \mid \varepsilon$
$T \rightarrow F T_R$
$T_R \rightarrow * F T_R \mid \varepsilon$
$F \rightarrow (E) \mid \mathbf{id}$

***