DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF BARISHAL

FINAL EXAMINATION-2022
Course Title: Compiler Design and Construction
Course Code: CSE-4103
4th Year 1st Semester; Admission Session: 2018-19

Time: 3 hours **Marks:** 60

Answer any Five Questions from the followings.

1. a) Analyze: Static versus Dynamic Checking. 4

   b) Predictive parsing is a special form of recursive descent parsing where we use one LOOKAHEAD token to unambiguously determine the parse operations. When a nonterminal has multiple productions, each production is implemented in a branch of a selection statement based on input look-ahead information. Execute the example (Fig. 1) using **Predictive Parser**. 8

Figure 1 →

type → *simple*
| ^ **id**
| **array** [*simple*] **of** *type*
simple → **integer**
| **char**
| **num dotdot num**

Input: **array [num dotdot num] of integer**

↑
lookahead

2. a) Define type equivalence. Explain the role of intermediate code generator in compilation process. 3

   b) Construct a quadruple, triples for the following expression: a +a*(b-c)+(b-c)*d? 5

   c) Define peephole optimization. Explain machine dependent and machine independent optimization? 4

3. a) Define tokens, patterns, and lexemes with examples. What is the function of the lexical analyzer? Demonstrate the Interactions between the lexical analyzer and the parser. 4

   b) What are operations on Language? Let *L* be the set of letters {A, B, …, Z, a, b, …, z} and let *D* be the set of digits {0, 1,…,9}. We may think of *L* and *D* in two, essentially equivalent, ways. One way is that *L* and *D* are, respectively, the alphabets of uppercase and lowercase letters and of digits. Show some other languages that can be constructed from languages *L* and *D*, using the union, concatenation, Kleene closure and Positive closure operators. 4

   c) Briefly explain the rules that define regular expressions over some alphabet and the languages that those expressions denote using basis and induction. How do you recognize the reserved words and identifiers? Show the transition diagram of **relop**. 4

4. a) What is a *dangling else*? 5

   b) How can the following grammar be ambiguous for ***(id - id / id)*** ? 7

   E → E-E | E/E | -E| (E) | id

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

b) Discuss about Handles. 2

c) The following is a substring of grammar symbols: 4

a b b c d e

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

Prove that this substring shows the property of "Handle".

6. a) Describe algorithm for LR Parser. 6

b) Based on LR Parsing, for the following grammar, show all steps for the stack, input, and action. 6

Grammar:
1. $E \rightarrow E + T$
2. $E \rightarrow T$
3. $T \rightarrow T * F$
4. $T \rightarrow F$
5. $F \rightarrow (\ E\ )$
6. $F \rightarrow \mathbf{id}$

7. a) Distinguish the following terms for error recovery in LR Parsing: 4
i) Panic mode,
ii) Phrase-level recovery

b) Construct the predictive parser for the following grammar 4
S->(L)/a
L->L,S/S

c) Construct the LR parsing table for the following grammar: 4
E→E + T | T
T→T * F | F
F →(E)|id

8. a) Let, you would like do parsing for syntax analyzer. Write down a table to show *FOLLOW(A)*. 6

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

b) Give specific examples of static checking for the followings: 6
a) Overloading, Coercion, and Polymorphism, b) Flow-of-Control Checks.

---***---