Date: April, 11th 2025

Course Instructors

Mr. Aamir Raheem, Dr. Faisal Aslam

Dr. Aatira Anum, Mr. Hamad ul Qudous

Total Time (Hrs):

Total Marks: 3

Total Questions:

Roll No   Section   Student Signature

Do not write below this line

Attempt all the questions.

CLO # 1: Describe the basic phases of Compilation and recognize the theories and techniques used in each phase

Q1) Remove left recursion from the following CFG. Also, perform left factoring where necessary.
[10 marks]

```
L -> L S | Halt

S -> T id ;

S -> T id () {}

T -> int | char
```

$A \rightarrow$

B

CLO # 3: Design and implement a simple lexical analyzer and Parser for a given CFG

Q2) Implement a recursive-descent parser in C++ for the following grammar: [10 mark

```
F -> T id ( P ) { )

T -> int | char

P -> T id ; P | ^
```

Here, ^ implies empty string (ε). Assume you are already provided with a function getToken(), wh returns the next token.

CamScanner

CLO # 1: Describe the basic phases of Compilation and recognize the theories and technique in each phase

Q3) Consider the following CFG for the C structures: [10

```
S -> struct id { L } ;        ∧1

L -> L T id ; | ^             ∧ 2
L -> ∧                        ∧2·1
T -> int | char              ∧ 3
T -> char                    ∧ 4
```

Here, ^ implies empty string (ε).

Now add actions to compute the size of a structure. For example, the size of the following st is 5 bytes (assuming an integer takes 4 bytes while a character takes one byte).

```
struct Student {

        int rollNo;

        char gender;

    };
```

After the computation, your translation scheme (SDT) shall print the final value. Do not change the grammar. Use local variables only!