Roll Number:_____

Thapar Institute of Engineering & Technology, Patiala
Department of Computer Science and Engineering

B. E. (Final Year) Auxillary Examination — Course code: UCS 802
Course Name:Compiler Construction
Time: 3 Hours, M. Marks: 100 — Name Of Faculty: Dr. Sunita Garhwal
DATED: 18/02/2020

Note: Attempt all questions. Assume missing data, if any, suitably

Q1. Consider the following grammar G_1: (15)

$$S \rightarrow AS \mid b$$
$$A \rightarrow SA \mid a$$

a) Construct the DFA of LR(0) items for grammar G_1
b) Construct LR(0) parsing table.
c) Show the parsing stack and the actions for the input string : **w= baab**.

Q2. Consider the following grammar G2 and perform the following(s) : (15)

$$S \rightarrow V = E$$
$$E \rightarrow F \mid E + F$$
$$F \rightarrow V \mid \textbf{integer} \mid (E)$$
$$V \rightarrow \textbf{id}$$

a) Construct the DFA of LALR(1) items.
b) Construct LALR(1) parsing table.
c) Show the processing of input string : **w=id=integer+id**.

Q3. Execute the following program and explain its output in terms of the runtime environment : (15)

```
int x=2

void f(int n) {
        static int x=1;
        g(n);
        x--;        }

 void g( int m)  {
        int y = m-1;
        if ( y>0 )  {
        f(y) ;
        x-- ;
                    }}

  main()     {
       g(x);
       return 0;   }
```

Q4. Explain the five phases of compiler. Illustrate with help of some example. (10)

Q5. Consider the following expression:
(x /y + z) * (y + z) –(x +y + z) (15)

a) Write sequence of three-address instructions that would be generated by above expression.
b) Represent the Quadruples, Triples and Indirect-Triple implementation for the above three-address code.

Q6. Consider the following grammar: (10)

$G \rightarrow Xa$

$X \rightarrow Gb \mid c$

a) Remove the left recursion.

b) Construct First and Follow sets for the non-terminals of the resulting grammar.

Q7. Explain the term Left factoring. Illustrate with help of some example (10)

Q8. Given the regular expression r = (a | b)*aba . Convert it into NFA using Thompson's Construction. Convert the obtained NFA into DFA and minimize it. (10)