

**322652(22)**

**B. E. (Sixth Semester) Examination, April-May 2020**

**(New Scheme)**

**(CSE Engg. Branch)**

**COMPILER DESIGN**

***Time Allowed : Three hours***

***Maximum Marks : 80***

***Minimum Pass Marks : 28***

***Note : Part (a) is compulsory for each question.  
Solve any two from (b), (c) and (d). All  
questions carry equal marks.***

**Unit-I**

1. (a) What is a compiler? 2
- (b) Explain the different phases of compiler with example. 7

[ 2 ]

- (c) Convert  $b(a+b)^*a$  into a DFA. 7
- (d) What is LEX? Explain with suitable code. 7

### Unit-II

2. (a) What do you mean by left recursion? Support your answer with example. 2
- (b) Give the rules to find out FIRST and FOLLOW of Non terminals of grammar. For the given grammar find out FIRST and follow : 7

$$E \rightarrow TE^1, E^1 \rightarrow +TE^1 | \epsilon, T \rightarrow FT^1,$$

$$T^1 \rightarrow *FT^1 | \epsilon, F \rightarrow (E) | id$$

- (c) Construct the predictive passes LL(1) table for the given grammar : 7

$$S \rightarrow AaAb | BbBa, A \rightarrow \epsilon, B \rightarrow \epsilon$$

- (d) Construct SLR table for the given grammar : 7

$$S \rightarrow AA, S \rightarrow aA, A \rightarrow b$$

### Unit-III

3. (a) What is role of semantic analyser? 2

[ 3 ]

- (b) Explain synthesized and Inherited attribute with example. 7

- (c) For the given grammar
- $$E \rightarrow E_1 + T, E \rightarrow E_1 - T, E \rightarrow T, T \rightarrow T_1 * F,$$
- $$T \rightarrow T_1 / F, T \rightarrow F, F \rightarrow (E), F \rightarrow \text{num}$$

Give the semantic rule to represent syntax directed definition for arithmetic expression and draw annotated parse tree for  $4 - 6/3 + 5$ . 7

- (d) For the given arithmetic expression,

$$Q := -b * (c + d)$$

give the quadruple, triple and indirect triple representation with brief explanation. 7

### Unit-IV

4. (a) Define local and global variable. 2
- (b) Explain about Activation Record. 7
- (c) Explain different allocation strategies. 7
- (d) Explain different parameters passing methods. 7

Unit-V

5. (a) Explain any two local optimization technique. 2

(b) Translate the following code to basic block and flow graph 7

```
Sum := 0
```

```
U := 1
```

```
While (i ≤ 10) do
```

```
  }
```

```
    Sum := sum + a [ 2 × i ]
```

```
    i := i + 1
```

```
  }
```

```
avg = sum / i
```

(c) Explain different issues related to code generation. 7

(d) Generate the code sequence for assignment

```
a := x - y + x - z + x - z. 7
```