



**School of Computer Science and Engineering**

**Fall Semester 2023-24 UG Seniors (B.Tech 2021 Batch)**

**Continuous Assessment Test – 2**

**SLOT: C1+TC1 & C2+TC2**

**Programme Name & Branch : B.Tech Computer Science and Engineering**

**Course Name & code: COMPILER DESIGN - BCSE307L**

**Class Number (s):**

**Faculty Name (s):**

**Exam Duration: 90 Min.**

**Maximum Marks: 50**

**General instruction(s):**

| Q.No. | Question   | Max Marks    | CO  | BL  |
|-------|--|--------------|-----|-----|
| 1.    | Consider the following grammar<br>$S \rightarrow S$ and $A A$<br>$A \rightarrow AB B$<br>$B \rightarrow B$ or $ 0 1$<br><br>Construct the SLR Parsing table for this grammar,<br>also Parse the input 1 or 0 and 1   | (5+3+2=10)   | CO2 | BL3 |
| 2.    | For the given grammar<br>$E \rightarrow SS$<br>$S \rightarrow xS$<br>$S \rightarrow y$<br><br>Construct set of LR(1) items for LALR Parser and<br>parse the input string $xyy$   | (5+3+2=10)   | CO2 | BL3 |
| 3.    | Construct a syntax directed translation scheme that<br>translates arithmetic expressions from postfix<br>notation into infix notation. Give annotated parse<br>tree for the expressions: $95-2^*$ and $952^*.-$ .  | (4+3+3 =10)  | CO2 | BL4 |
| 4.    | Generate the Quadruple, Triple and Indirect Triple<br>representation of the three address code of the<br>following statement $a + b * c / e \uparrow f + b * a$  | (4+2+2+2=10) | CO2 | BL4 |
| 5.    | Generate an intermediate code for the following<br>expression<br>$E=m+n$<br>$For(i=0;i<10;i++)$<br>$For(j=0;j<10;j++)$<br>$C=a[i]+b[j];$<br>$Print c;$<br><i>Do we need back patching in the above case ?if so</i><br><i>Why back patching is essential?</i> | 7+3          | CO2 | BL4 |