**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | JUNE 2024 | Maximum Marks | 50 |
| Course Code | 21CS63 | Duration | 90 Min |
| Sem | VI Semester | Test-I | |
| **COMPILER DESIGN**  **(Common to CS, IS)** | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** |  | **M** | **BT** | **CO** |
| 1 | Consider the statement:  A=b\*c\*d+50;  Show the input and output of each phase of the compiler with relevant explanation.  **Solution: explanation on all the 6 phases = 6 marks**  **Symbol table = 1 mark**  **Input and output for each phase= 3 marks** | 10 | **2** | 1 |
| 2 a | With a neat diagram explain the role of Parser. Justify why tokenization and error checking is divided into two phases of the compiler.    Diagram + explanation = 1+3 = 4marks  Justification:  2Marks | 6 | 2 | **2** |
| 2b. | List and explain the different error recovery strategies used in syntax analyzers.  **Panic mode Recovery**  **Phrase level error recover**  **Error productions**  **Global Corrections 4\*1=4Marks** | 4 | 2 | **2** |
| 3 | Consider the grammar given below:  **E → T + E | T**  **T → float | float \* T | (E)**   1. Make necessary changes to make it suitable for LL(1) parsing. 2. Construct FIRST and FOLLOW sets. 3. Construct the predictive parsing table.   Show the moves made by the predictive parser on the input )float\*\*float+.  **a) Removal of left factoring – 2Marks**  **b) First and Follow set calculation – 3Marks**  **c) Construction of parsing table – 2 Marks**  **d) Parsing the input – 3 Marks** | 10 | **4** | **3** |
| 4. | For each of the following grammar:  **a) S🡪 SAT | TB | s**  **T🡪aTb | ab**  **A🡪 Aa | a**  **B🡪Bb | b**  **b)** **S🡪 ASA | aB**  **A🡪 BA | BS | a**  **B🡪b | ε**  Perform the following  1.Modify the following grammar to suit the requirements of Top Down parser  2.Find the FIRST and FOLLOW set for the grammar | **10** | 3 | 3 |
| 5a. | Write the rules for LL(1) grammar and show that the following grammar is LL(1) by constructing the Top own parsing table:  S 🡪 AaAb | BbBa  A🡪  **ε**  B🡪  **ε**  **2 Rules – 2 Marks**  **Fist set and Follow set calculation – 4 marks**  **Construction of the table—4 Marks** | 5 | **4** | 3 |
| 5b. | Define token, lexeme and pattern and identify the same in the following C statement:  **int a;**  **for( a=1;a<10;a++)**  **printf(“RVCE”);**  Definition token, lexeme and pattern – 2 Mark   |  |  |  | | --- | --- | --- | | lexeme | pattern | token | | int, for | int, for | <int> <for> | | a, printf | letter followed by letter /digit | <id> | | 1,10 | 0 to 9 | <num> | | ( ) ; , | punctuation symbols | <)> <,> <;> <)> | | = < ++ | operators | <=> << > <++> | | “RVCE” | Anything surrounded by “ “ | Literal |   0.5\*6= 3 Marks | 5 | **3** | 4 |

**COURSE OUTCOMES**

|  |  |
| --- | --- |
| **CO1:** | Understand and explore the fundamental concepts of compiler design and its implementation.. |
| **CO2:** | Identify and apply rules for designing various phases of compiler Analyse a problem and  identify the suitable data structure to develop solution |
| **CO3:** | Analyse the practices adopted in constructing an efficient compiler. |
| **CO4:** | Implement and demonstrate in-depth knowledge of various technologies related to principles, techniques  and tools for designing compiler. |

BT-Blooms Taxonomy, CO-Course Outcomes, M-Marks

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Marks Distribution | Particulars | | CO1 | CO2 | CO3 | CO4 | L1 | L2 | L3 | L4 | L5 | L6 |
| Test | Max Marks | 10 | 10 | 25 | 05 | -- | 20 | 15 | 15 | - | - |