**18CS602**

**Hall Ticket Number:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **III/IV B.Tech (Regular/Supplementary) DEGREE EXAMINATION** | | | |
| **JUNE, 2022** | **Computer Science and Engineering** | | |
| **Sixth Semester** | **Compiler Design** | | |
| **Time:** Three Hours | | **Maximum :** 50 Marks | |
| *Answer Question No.1 compulsorily.* | | | (1X10 = 10 Marks) |
| *Answer ONE question from each unit.* | | | (4X10=40 Marks) |

**1.** Answer all questions (1X10=10 Marks)

|  |  |
| --- | --- |
| a | Define Compiler. |
| b | Specify the structure of the Lex Program. |
| c | What is left-factoring? |
| d | What is an LR(0) item (item for short) of a grammar G? |
| e | What is Yacc? |
| f | Differentiate Syntax tree and DAG? |
| g | What is Back patching? |
| h | Define Basic block. |
| i | In programming language semantics, the term *environment*refers \_\_\_\_? |
| j | Define symbol table. |
|  |  |

**UNIT – I**

|  |  |  |
| --- | --- | --- |
| 2. | Explain phases of the compiler with neat sketch for the following statement.  position := initial + rate \* 60 | 10 M |

**(OR)**

|  |  |  |
| --- | --- | --- |
| 3.a | Explain input buffering in detail. | 5 M |
| 3.b | Construct predictive parsing table for the following grammar.  *E → E + T | T*  *T → T \* F | F*  *F → ( E ) | id* | 5 M |

**UNIT – II**

|  |  |  |
| --- | --- | --- |
| 4. | Construct SLR parsing table for the following grammar. | 10 M |

**(OR)**

|  |  |  |
| --- | --- | --- |
| 5.a | Explain Synthesized and Inherited attributes with example. | 5 M |
| 5.b | Construct syntax tree, dag for the statement a\*(b+c)+(b+c). | 5 M |

**UNIT – III**

|  |  |  |
| --- | --- | --- |
| 6.a | Write three address statements and give quadruple, triple and indirect triples representations to the statement a = b\*(-c) + b\* (-c). | 6 M |
| 6.b | Explain different types of intermediate representations. | 4 M |

**(OR)**

|  |  |  |
| --- | --- | --- |
| 7.a | Explain code generation algorithm with function getreg(). | 6 M |
| 7.b | Generate target code using the code generation algorithm for the following example.  t = a - b  u = a - c  v = t + u  d = u + v | 4 M |

**UNIT – IV**

|  |  |  |
| --- | --- | --- |
| 8.a | Explain following storage allocation strategies in detail.   1. Static allocation strategy 2. Stack allocation strategy | 6 M |
| 8.b | What is an activation record? Specify the fields of an activation record. | 4 M |

**(OR)**

|  |  |  |
| --- | --- | --- |
| 9.a | Explain different data structures to implement symbol table. | 6 M |
| 9.b | Write a short notes on symbol table entries. | 4 M |

****