Name: ____________________

Reg. No.: □□□□□□□□□□

END-SEMESTER EXAMINATION (DEC– 2017)

SEMESTER – V (SESSION – 2017-2018)

(B.Tech. CSE)

Subject Code: CS0301
Subject: Compiler Design

Duration: 3 hours
Max. Marks: 100

Instructions

- **All Questions are compulsory**
- **The Question paper consists of 2 sections - Part A contains 10 questions of 2 marks each. Part B consists of 5 questions of 16 marks each.**
- **There is no overall choice. Only Part B question include internal choice.**

PART – A

*(2 * 10 = 20 Marks)*

1. Is macro processing a phase in compilation? Justify your answer.
2. What is YACC?
3. What are the different data structures used for symbol table?
4. Define l-value and R-value.
5. List the properties of LR parser.
6. Define parser. Mention basic issues in parsing.
7. Construct a DAG for the expression a=b*-c + b*-c.
8. Why we need Intermediate code generator.
9. What is syntax directed definition?
10. List the properties for regular expression.

PART – B

(16 * 5 = 80 Marks)

11. Explain the steps involved in a one-pass assembler with their block diagram and data structure used in it, and differentiate between pass one assembler and pass two assembler in detail.

OR

Explain with a small example how linking and loading of a required module of a program can be done depending on the runtime conditions during the program's execution.

12. Describe the various phases of a compiler in detail and trace the output of each phase for the program segment,

Position = *initial* + rate *60,

Where rate is a real data type

OR

a) Define lexeme, token and pattern. Identify the lexemes that make up the tokens in the following program segment. Indicate corresponding token and pattern.

```
void swap (int i, int j)
{
int t;
t=i;
I =j;
j=t;
}
```

b) Construct minimum state DFA for the following regular expression

(a |b)* a (a | b) (10 Marks)

13.Explain Code Generator design issues in detail and Translate a=b*-c+b*-c into three address statements, quadruples, triples and postfix form.

OR

Explain Optimization of basic blocks and Data flow analysis of structural programs with example.

14. What is the use of a symbol table? How identifiers are stored in the symbol table also explain the different storage allccation strategies?

OR

a) Explain syntax directed definition of a simple desk calculator. Using that draw annotated parse tree for

3 * 5 + 4n (8)

b) Define Type Checker. Write down the specification of a simple Type Checker. (8)

15. Differentiate between recursive descent parsing and Predictive Parsing and find out First and Follow of given grammar.

S → i E t S S' | a
S' → e S | ε
E → b

Construct a predictive parsing table and check whether the following grammar is LL(1) or not a grammar.

OR

Explain LR Parsing and its type with proper example.