CS61/CS1561

RAMAIAH
Institute of Technology

USN	1	M	S							

(Autonomous Institute, Affiliated to VTU)
(Approved by AICTE, New Delhi & Govt. of Karnataka) Accredited by NBA & NAAC with 'A' Grade

SEMESTER END EXAMINATIONS – JUNE 2019

Course & Branch	:	**B.E. : Computer Science and Engineering**	**Semester**	:	**VI**
Subject	:	**Compiler Design**	**Max. Marks**	:	**100**
Subject Code	:	**CS61/CS1561**	**Duration**	:	**3 Hrs**

Instructions to the Candidates:

- Answer one full question from each unit.

UNIT- I

1. a) What are the phases under analysis part of compiler? Show their operations with an example. CO1 (08)

b) Justify the purposes of input buffers in compiler operations. CO1 (08)

c) How are panic mode and phrase level syntax error recovery schemes work? CO1 (04)

2. a) How the parser interacts with the Lexer? Explain. CO1 (08)

b) Define the following with proper examples: CO1 (08)
i) Regular expressions ii) Languages.

c) Show ambiguity in grammar with an example grammar and a proper input string. CO1 (04)

UNIT- II

3. a) Compute the FIRST and FOLLOW of all the Non-terminals for the grammar CO2 (12)

S□L=R | R
L□*R | id
R□ L

Is this Grammar suitable for Predictive Parsing? If not make necessary changes and construct a LL (1) parsing table.

b) Construct a SLR parsing table for the grammar given below. Check whether the grammar is SLR or not. CO2 (08)

S→ AaAb | BbBa
A→ ε
B→ ε

4. a) Construct LALR(1) set of items for the grammar G: CO2 (10)

G: S□ Aa | bAc | d | bda
A□d

b) Show the moves made by the stack of a shift reduce parser for accepting the input " id+id+id" CO2 (06)

G: E□E+T | T
T□id

c) Define handle. Identify handles for the given input string : aabb CO2 (04)

G: S□ aSb | ab

UNIT- III

5. a) Generate the SDT for type setting boxes for the following grammar: CO3 (08)

    S→B
    B→BB | B sub B | text

   b) Explain how the side effects are controlled in simple type declarations by defining the SDD. CO3 (08)

    D□T L
    T□int
    L□ L, id | id

   c) Write the rules for turning L-Attributed Definitions to SDT. CO3 (04)

6. a) Distinguish between the terms: CO3 (06)
    i) Inherited attribute and Synthesized attribute
    ii) Annotated Parse Tree and attribute grammar
    iii) S-Attributed Definition and L- Attributed Definition.

   b) Illustrate how the desk calculator is implemented on a bottom-up parsing stack with the help of semantic actions. CO3 (08)

   c) Draw the dependency graph for the following expressions: CO3 (06)
    i) int a, b, c ii) char a, b, c, d

UNIT- IV

7. a) Construct DAG and VNM array for the following expressions. CO4 (08)
    i) (a+a)*(a+(a+a)*a+a)
    ii) (d+f+(g+h)*(d+f))

   b) When are two type expressions equivalent? Explain. CO4 (04)

   c) Show the SDD for flow of control statements like if…else, while. CO4 (08)

8. a) Write a C program segment to add two arrays using a loop. Write three address code for your program segment. Represent the same in quadruples, triples and indirect triples. CO4 (08)

   b) Write SDD to generate three address code incrementally for arithmetic expressions. CO4 (08)

   c) Write the Syntax Directed Translation for switch statement. CO4 (04)

UNIT- V

9. a) List and Explain the issues in the development of a code generator. CO5 (10)

   b) Describe about the several code improving transformations on the code represented by the basic block. CO5 (06)

   c) Construct the DAG for the following basic block: CO5 (04)

```
 x=a[i]
a[j]=y
z=a[i]
```

10. a) Define Basic blocks. Give an algorithm to partition three address instructions into basic block. CO5 (08)

    b) Convert the following program fragment into **three address code** and obtain CO5 (12)
    i) Basic Blocks
    ii) Flow graph

```
Code: int a[10][10];
        i=0; j=0;
        while(i<10){
              a[i][j]=1;
              i=i+1;
              j=j+1;
                    }
```

****