VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of the UGC Act, 1956)

Reg. No. :

Final Assessment Test(FAT) - Nov/Dec 2024

Programme	**B.Tech.**	Semester	**Fall Semester 2024-25**
Course Code	**BCSE307L**	Faculty Name	**Prof. Sureshkumar Wi**
Course Title	**Compiler Design**	Slot	**G1+TG1**
		Class Nbr	**CH2024250101291**
Time	**3 hours**	Max. Marks	**100**

General Instructions

- Write only Register Number in the Question Paper where space is provided (right-side at the top) & do not write any other details.

Course Outcomes

1. Apply the skills on devising, selecting, and using tools and techniques towards compiler design
2. Develop language specifications using context free grammars (CFG).
3. Apply the ideas, the techniques, and the knowledge acquired for the purpose of developing software systems.
4. Constructing symbol tables and generating intermediate code.
5. Obtain insights on compiler optimization and code generation.

Section - I
Answer all Questions (10 × 10 Marks)

*M - Marks

Q.No	Question	*M	CO	BL
01.	i. Show how the given high level language statement *int result = (value1+value2)–value3*1.4* is converted to machine code by illustrating the outcome at each stage of compilation process. (3 Marks) ii. Construct a minimized Deterministic Finite Automata for the given regular expression using direct method: $(a\|b)^*c(a\|b\|c)^+$ (7 Marks)	10	1	3
02.	i. For the following grammar, devise predictive parser and show the parsing table. (7.5 Marks) S → X X → Yy | Za Y → xY | ε Z → zZ | ε ii. Show the moves of the parser on the given input string: xxxxxy (2.5 Marks) Note: S, X, Y and Z are non-terminals. a, x, y and z are terminals.	10	2	3
03.	Consider the following augmented grammar in which S', S, J and K are non-terminals. a, b, c, d and f are terminals. S'→S S→aJKf J→Jbc | b K→d Build Canonical LR parser by constructing the following: i. LR(1) Item set (4 Marks)	10	3	3

ii. Parser Table (6 Marks)

04. Consider the following SDD: 10 2 3

Production	Semantic Rule
T→FG	G.inh=F.val T.val=G.syn
G→*FG$_1$	G$_1$.inh=G.inh × F.val G.syn= G$_1$.syn
G→ε	G.syn=G.inh
F→digit	F.val=digit.lexval

i. Construct an annotated parse tree for the expression *1*2*3*4* (5 Marks)
ii. Draw the corresponding dependency graph to show the order of evaluation of attribute values. (5 Marks)

05. Translate the given segment of code into the following intermediate code representations: 10 4 3
i. Triple (3 Marks)
ii. Indirect Triple (3 Marks)
iii. Quadruple (4 Marks)
Code:

```
if((a+b)>c || d==e)
        res=res+10;
else
        res=res-10;
fin=a+b+c+d+e;
```

06. Consider the following Boolean expression and illustrate the working of backpatching: 10 4 3
(a<b || c<d) && (e>f || b>d) && (a+c!=b+d)
i. Generate productions and semantic actions for Boolean Expression (4 Marks)
ii. Construct parse tree (3 Marks)
iii. Give the three address code using backpatching (3 Marks)

07. i. Consider the following code snippet: 10 5 4

```
a=a*0;
b=b+0;
for(i=10;i<15;i++)
{
a=a+5;
c=10;
}
for(i=10;i<15;i++)
{
c=c*2;
b=0;
}
```

Apply suitable optimization techniques to this code with necessary explanation and write the final optimized code. (5 Marks)
ii. Draw syntax tree and DAG for the following statement: (5 Marks)
a=((a+b)-((a-b)*(a-b)))+((a-b)*(a-b))

08. For the given code segment, construct the following: 10 5 3
i. Three address code (4 Marks)
ii. Basic blocks (3 Marks)
iii. Control flow graph (3 Marks)
Code:

```
for(i = 1 to n)
{
counter = 3;
b=b+d-e;
```

```
while (counter > 0)
{
a=b+c*d;
counter--;
}
}
```

09. Generate a typical machine code for the given code by assuming 'a' and 'b' are arrays whose elements are 4-byte values. (6 Marks) 10 5 4
For your machine code, determine its total cost of all instructions and usage counts of all variables. (2 Marks + 2 Marks)
Code:

```
sum=0;
i=0;
L1: if i>n goto L2;
sum=sum+a[i];
sum=sum+b[i];
i=i+1;
Goto L1;
L2:
```

Note: Develop a machine code by assuming that, if a machine instruction contains two operands, first operand represents destination and second operand represents source.

10. In the following machine code, R1, R2, R3 and R4 represent CPU registers. a and b represent memory locations. First operand is a destination and second operand is a source. 10 5 4

```
LOAD R1, a
LOAD R2, b
MOV R3, R1
SUB R1, R2
MOV R4, R1
MOV R1, R3
ADD R1, R2
STORE a, R4
STORE b, R1
```

i. For the given machine code, draw the data dependency graph. (4 Marks)
ii. For the given machine code, assume that the MOV instruction consumes 1 clock cycle, ADD/SUB instructions consume 2 clock cycles and LOAD/STORE instructions consume 3 clock cycles. Consider the three machines 1, 2 and 3 with following instance of resources and find the shortest schedule for each machine to achieve maximum parallelism. (6 Marks)
Machine 1: one ALU resource and one MEM resource
Machine 2: one ALU resource and two MEM resources
Machine 3: two ALU resources and one MEM resource

BL-Bloom's Taxonomy Levels - (1.Remembering, 2.Understanding, 3.Applying, 4.Analysing, 5.Evaluating, 6.Creating)