



**COLLEGE OF ENGINEERING, PUNE**  
(An Autonomous Institute of Government of Maharashtra.)

**END Semester Examination**

**Programme: B.Tech**

**Semester: VII**

**Course Code: IT(DE)-18008**

**Course Name: DE-III Language Processor**

**Branch: Information Technology.**

**Academic Year: 2018 - 19**

**Duration: 3 Hrs**

**Max Marks: 60**

**Student MIS No.**

1	1	1	5	0	8	0	6	5
---	---	---	---	---	---	---	---	---

**Instructions:**

1. **Question no 1 carries negative mark**, figures to the right indicate the full marks.
2. Mobile phones and programmable calculators are strictly prohibited.
3. Writing anything on question paper is not allowed.
4. Exchange/Sharing of stationery, calculator etc. not allowed.
5. Write your MIS Number on Question Paper.

- Q 1 (A) 1) For the C program given below the statement that does not hold true is 12
- ```
for (i = 0; i for(j = 0; j if (i%c)
{
x += (4*j+5*i);
y+=(7+4*j);
}}}
```
- a. There is a scope strength reduction  
☒ b. There is a scope of dead code elimination  
c. There is a scope of common sub-expression elimination  
d. None of the above
- 2) Compiler translates the source code to
- a. Executable code  
b. Machine code  
c. Binary code  
d. Both b & c
- ☒ 3) In compilers generation of intermediate code based on an abstract machine model is useful because
- a. Syntax-directed translations can be written for intermediate code generation  
b. To generate code for real machines directly from high-level language programs is not possible  
c. Portability of the front end of the compiler is enhanced  
d. Implementation of lexical and syntax analysis is easier
- 4) Consider the grammar where P, Q, R are not terminals and r, s, t are terminals
- a.  $P \rightarrow Q R$   
b.  $P \rightarrow Q s R$   
c.  $P \rightarrow \epsilon$   
d.  $P \rightarrow Q t R r$
- The grammar rules that violate the requirements of an operator grammar is



**COLLEGE OF ENGINEERING, PUNE**  
(An Autonomous Institute of Government of Maharashtra.)

- ☒ a. a and c only
- b. b and c only
- c. a and d only
- d. a only

5) Which one of the following statement is false for the SLR (1) and LALR (1) parsing tables for a context free grammar?

- a. The reduce entries in both the tables may be different
- b. The error entries in both the tables may be different
- ☒ c. The go to part of both tables may be different
- d. The shift entries in both the tables may be identical

6) When is the type checking usually done?

- ☒ a. During syntax directed translation
- b. During lexical analysis
- c. During code optimization
- d. During syntax analysis

DL 7) What information need to be included in an object module, if a linker is given object modules for a set of programs that were compiled separately?

- a. Names and locations of all external symbols defined in the object module
- b. Object code
- c. Absolute addresses of internal symbols
- d. Relocation bits

8) In some programming language, L denotes the set of letters and D denotes the set of digits. An identifier is permitted to be a letter followed by any number of letters or digits. The expression that defines an identifier is

- a.  $(L.D)^*$
- b.  $(L + D)^*$
- c.  $L(L.D)$
- ☒ d.  $L(L + D)^*$

9) Which one of the following statement is true?

- a. Canonical LR parser is more powerful than LALR parser
- ☒ b. SLR parser is more powerful than LALR
- ☒ c. LALR parser is more powerful than canonical LR parser
- ☒ d. SLR parser, canonical LR parser and LALR parser all have the same power

10) How many tokens are there in the following C statement?

`printf("j=%d, &j=%x", j&j)`

- a. 11
- b. 12
- ☒ c. 9
- d. 10

11) To convert an arbitrary CFG to an LL (1) grammar

- a. Factor the grammar alone
- b. Remove left recursion alone





# COLLEGE OF ENGINEERING, PUNE

(An Autonomous Institute of Government of Maharashtra.)

- c. Remove left recursion and factor the grammar  
d. None of the above

C inf

- 12) Why is the code optimizations are carried out on the intermediate code?
- Because for optimization information from the front end cannot be used
  - Because program is more accurately analyzed on intermediate code than on machine code.
  - Because for optimization information from data flow analysis cannot be used
  - Because they enhance the portability of the compiler to the other target processor.

Q 2 (A) E=Given the grammar  
S → aABe  
A → Abc  
A → b  
B → d

LR (a)

04

Deduce the sets items.

(B) Write a lex code to recognize following tokens of the C language (Any three )

02

- Comment \\*.....\*\
- Logical operators: &&, ||, !
- Identifiers
- Float numbers

(C) How lexical analysis detect the errors? Explain with suitable example

06

Or

Write Syntax Directed Translation scheme for boolean expression. Explain use of backpatching.

Q 3 (A) Show MNT, MDT entries and output file after the following shown file undergoes a macro expansion.

04

```
MACRO      EVAL &X, &Y
            LOAD &X
            ADD  &Y
            STORE &X

MEND
MACRO      MAJOR &A, &B, &C
            EVAL  &A, &B
            EVAL  &A, &C

MEND

            LOAD D
            LOAD A
            LOAD G
            MAJOR D,A,G

END
```

(B) Elaborate the use of stack data structure in expansion of nested macro calls.

OR

04

An assembly language program performs certain task at 10 places .under what conditions would you code this action as

- A Macro ?
- A Subroutine ?

Justify your answer with the help of appropriate example



**COLLEGE OF ENGINEERING, PUNE**  
(An Autonomous Institute of Government of Maharashtra.)

- (C) Write the advantage and disadvantage of an absolute Loader 04
- Q4 (A) Given the grammar 04

$S \rightarrow L = R$   
 $S \rightarrow R$   
 $L \rightarrow *R$   
 $L \rightarrow id$   
 $R \rightarrow id,$

form its SLR parsing table.

- (B) Consider the following grammar and test whether the grammar Is LL(1) or not 04

$S \rightarrow lAB \mid \epsilon$

$A \rightarrow lAC \mid 0C$

$B \rightarrow 0S$

$C \rightarrow l$

- (C) What is S-attributed definition and L-attributed definition ? 04  
How conversion of L-attributed to S-attributed is done ?

- Q5 (A) Optimise following code by applying suitable code optimization technique 04

1) B1:  
a = 2 \* c  
goto B3  
B2:  
b = a  
B3:  
b = 33  
for ( i=0; i<n; i++)  
{  
Z = x+5;  
-----  
-----  
Y = x+5\*60;  
}

- (B) (i) Minimize the number of temporary variables for the following piece of code. The temporary variables are indicated as  $t_i$  04

$t_1 = a * a$   
 $t_2 = a * b$   
 $t_3 = 2 * t_2$   
 $t_4 = t_1 + t_3$   
 $t_5 = b * b$   
 $t_6 = t_4 * t_5$

- (ii) Generate intermediate code for the statement  $X[i] := Y + Z$ .

- (C) What are different issues in code generation ?

OR

What is peephole optimization