

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
HYDERABAD CAMPUS**

FIRST SEMESTER 2019 – 2020

PRINCIPLES OF PROGRAMMING LANGUAGES (CS F301) – MID SEM EXAM

Date: 3/10/2019

Weightage: 25% [50 Marks]

Duration: 90mins

Type: Closed Book

Please note: 1. All parts of the questions have to be answered consecutively.

2. Your answers should be brief.

Q1. Features of a Programming Language

- A. Given the following input string in C `12+a+++==42pop17-1+fif` what sequence of <lexeme, token> pairs are returned by the lexical analyzer. [3 M]

M]

[Note: The symbol == is double equal to]

- B. Suppose C language does not have / (division) operator [4+1=5 M]

- i. Write a program to divide one from other without using / operator.
- ii. What is the effect of not having / operator on readability and writeability of the language?

[Note: The program should imitate exactly '/' operator.]

Q2. Syntax of programming languages

- A. The following is the description of a valid identifier in Pascal: The identifier must start with a letter and be followed by zero or more letters, digits or underscores. An underscore cannot be the final symbol or be followed by another underscore. Give a regular expression to accept valid identifiers in Pascal. [2 M]

- B. Assume that you are hired by casio calculator company. You are assigned challenging tasks listed in questions i-viii. The BNF grammar of the original calculator is given below.

<PROGRAM>:: = <STATEMENT> | _____ [2+2+3+3+3+2+4+3=22 M]

M]

<STATEMENT> :: = <EXP> | id = <EXP> | unset id | list | quit | exit

<EXP> :: = <EXP> + <TERM> | <EXP> - <TERM> | <TERM>

<TERM> :: = <TERM> * <FACTOR> | <TERM> / <FACTOR> | <FACTOR>

<FACTOR> :: = id | number | (<EXP>) | sqrt (<EXP>)

- i. Fill the missing production for the non-terminal <PROGRAM> to accept multiple statements.
- ii. Add a new operator ** to compute power of a number/identifier which has higher precedence over +, -, * and / operators but must have lesser precedence than parenthesis and sqrt function.
- iii. Give an ambiguous grammar to add a new function **max()** which has same precedence as sqrt. The **max()** function computes the maximum value of one or more expressions. For example, **max(7+2)** would evaluate to 9, and **max(17, 6-42, 5)** would evaluate to 17.
- iv. Given the input string **max(17, 6-42, 5)** prove your solution for question iii is ambiguous.
- v. Modify the grammar and give unambiguous solution to question iii.
- vi. If we would like to add this **max()** function as provided in solution v, what changes, if any, need to be made to the original calculator during the lexical analysis phase? Describe any necessary changes to the lexer, the set of tokens, or anything else that must be modified.

[Note: You do not need to give the implementation, just describe the needed changes.]

- vii. Give an attribute grammar to implement **max()** function.

- viii. Draw the annotated/decorated parse tree to show how the values are synthesized for the input string **max(17, 6-42, 5)**

Q3. Data Types

[5+1=6M]

Shristy is working on designing an efficient way of storing a symmetric 2D matrix of size N×N. Since the array is symmetric she wants to store the upper triangular matrix including the principal diagonal elements using row major ordering. The following is an example of the array. The elements stored in indices that are shown in bold will be stored using row major ordering.

1	2	3	4	<p>A. Is it possible to formulate the effective address calculation to access each element of the array $A[i][j]$ for Shristy? If so, how can it be done, If not why?</p> <p>B. Show the runtime descriptor that will be generated for storing this array.</p>
5	6	7	3	
8	9	6	2	
10	8	5	1	

Q4. Type systems

[3+3+3=9 M]

A. For the input expression *a or b and c*

- Give the required type rules for Boolean operators (or, and) assuming that $a : \text{bool}$ $b : \text{bool}$ $c : \text{bool}$
- Construct a derivation tree to give a type for the input expression.

B. Give an example code for creating memory leak in Java.

Q5. Names Type Binding

[3

M]

Given the following code in Java show the memory snapshot when the statement *return 100* is being executed.

```
class Popl {
    public static void main(String[] args) {
        int i = 42;
        int[] a = new int[3];
        int k;
        k = test(i,a);
    }
    public int test(int m, int[] n) {
        n[0] = m;
        return 100;
    }
}
```


That's all folks