

COMPUTER SCIENCE

Paper – 1

(THEORY)

Three hours

*(Candidates are allowed additional 15 minutes for **only** reading the paper.
They must NOT start writing during this time)*

Answer **all** questions in Part I (compulsory) and **seven** questions from Part-II, choosing **three** questions from Section-A, **two** from Section-B and **two** from Section-C.

All working, including rough work, should be done on the same sheet as the rest of the answer.

The intended marks for questions or parts of questions are given in brackets [].

PART I

Answer **all** questions

While answering questions in this Part, indicate briefly your working and reasoning, wherever required.

Question 1

- (a) Verify using the truth table, $A + A = A$. Also state the law. [2]
- (b) If $X \Rightarrow Y$ then write its: [2]
- (i) Converse
- (ii) Contra positive
- (c) Show how a NAND gate can be used to construct an OR gate. [2]
- (d) Given, $F(X, Y, Z) = X.Y + \bar{Z}$ write the function in canonical sum-of-product form. [2]
- (e) Given, the Boolean Function, $F(X, Y, Z) = \Sigma(2, 3, 4, 6, 7)$. [2]
Reduce it using Karnaugh's Map. And also find the complement of its result.

Question 2

- (a) State the difference between linear and non-linear data structures. [2]
- (b) What is an Abstract class? [2]
- (c) A matrix $B[10][7]$ is stored in the memory with each element requiring 2 bytes of storage. If the base address at $B[X][1]$ is 1012 and the address at $B[7][3]$ is 1060, determine the value 'X' where the matrix is stored in **Column Major** form. [2]
- (d) What do you mean by Dominant term in Complexity? Give an example? [2]
- (e) Convert the following infix notation to its postfix form: [2]
 $(A + B) * (B * C) / (C - D * B)$

This Paper consists of 9 printed pages and 1 blank page.

Question 3

- (a) Give output of the following function where **a** and **b** are arguments greater than 0. Show the dry run/working.

```
int somefun (int a, int b)
{
    int an, s, k;
    if(a<b) {
        s = a; k = b;
    }
    else {
        s = b; k = a;
    }
    an = k;
    while(an % s != 0)
        an + = k;
    return an;
}
```

- (i) What will the function **somefun (4 , 7)** return ? [2]
(ii) What will the function **somefun (7 , 5)** return ? [2]
(iii) In one line, state what the function is doing, apart from recursion. [1]

- (b) The following function is a part of some class which searches for a value (search) in an array **arr[]** using the **Binary Search** technique. (Assume that the array arr[] has been sorted in ascending order. It returns the value 1 when the search is successful, otherwise it returns 0.

There are some places in the code marked by ?1?, ?2?, ?3?, ?4?, ?5? which must be replaced by a statement / expression so that the function works properly:

```
int binSearch (int arr[], int search)
{
    int lb = 0, ub = arr.length-1, mid, flag=0;
    while ( ? 1 ? ) {
        mid = ? 2 ?;
        if (arr[mid] == search)
        {
            ? 3 ?;
            break;
        }
        else if (search>arr[mid])
            ? 4 ?;
        else
            ? 5 ?;
    } //end of while
    if ( flag == 1 )
        return 1;
    else
        return 0;
}
```

- (i) What is the expression/value at ? 1 ? [1]
(ii) What is the expression/value at ? 2 ? [1]
(iii) What is the expression/value at ? 3 ? [1]
(iv) What is the expression/value at ? 4 ? [1]
(v) What is the expression/value at ? 5 ? [1]

PART – II

Answer seven questions in this part, choosing three questions from Section A, two from Section B and two from Section C.

SECTION - A

Answer any three questions

Question 4

- (a) Given the Boolean function: $F(A,B,C,D) = \Sigma (3, 4, 6, 7, 11, 12, 14, 15)$
- (i) Reduce the above expression by using 4 - variable K-Map , showing the various groups (i.e; octal , quads and pairs). **[4]**
- (ii) Draw the Logic gate diagram of the reduced expression. Assume that the variable and their complements are available as inputs. **[1]**
- (b) Given the Boolean function: $F(A,B,C,D) = \pi (0, 8, 10, 12, 13, 14, 15)$
- (i) Reduce the above expression by using 4 - variable K-Map , showing the various groups (i.e; octal , quads and pairs). **[4]**
- (ii) Draw the Logic gate diagram of the reduced expression. Assume that the variable and their complements are available as inputs. **[1]**

Question 5

A man goes to a matrimonial office to search for a bride. The office has information about 10,000 girls. The man gives two conditions:-

- The girl must be a teacher or graduate.
- The girl must be 25 years or less and beautiful.

THE INPUTS ARE:

G - The girl is a graduate (1 indicates yes and 0 indicates no)

T - The girl is a teacher (1 indicates yes and 0 indicates no)

A - The girl is 25 years or less (1 indicates yes and 0 indicates no)

B - The girl is beautiful (1 indicates yes and 0 indicates no)

OUTPUT IS:

S : The girl is selected [1 indicates she is selected and 0 indicates she is rejected]

- (a) Draw the truth table for the inputs and outputs given above and write the **POS** expression for S (G, T, A, B). **[5]**
- (b) Reduce S (G, T, A, B) using Karnaugh's map. **[5]**

Draw the logic gate diagram for the reduced **POS** expression for S (G, T, A, B).

You may use gates with two or more inputs. Assume that variable and their complements are available as inputs.

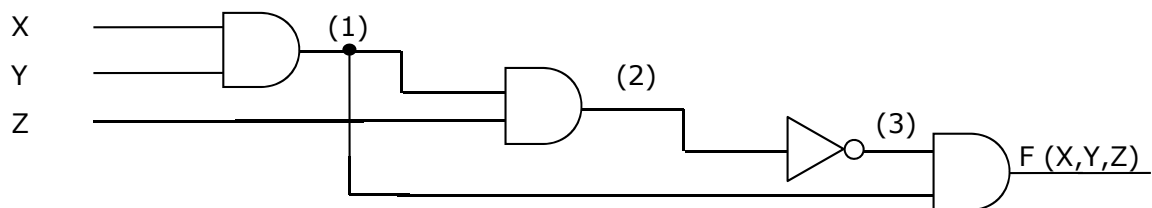
Question 6

- (a) In the following truth table X and Y are inputs and B and D are outputs: **[3]**

X	Y	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Answer the following questions:

- (i) Write the SOP expression for D.
 - (ii) Write the POS expression for B.
 - (iii) Draw the logic diagram for SOP expression derived for D, using only NAND gates.
- (b) Using truth table verify: $(\sim p \Rightarrow q) \wedge p = (p \wedge \sim q) \vee (p \wedge q)$. **[3]**
- (c) From the logic circuit diagram given below, name the outputs (1), (2) and (3). Finally derive the Boolean expression and simplify it to show that it represents a logic gate. Name and draw the logic gate. **[4]**



Question 7

- (a) A combinational logic circuit with three inputs P, Q, R produces output 1 if and only if an odd number of 0's are inputs. **[4]**
- (i) Draw its truth table
 - (ii) Derive a canonical SOP expression for the above truth table.
 - (iii) Find the complement of the above derived expression using De Morgan's theorem and verify if it is equivalent to its POS expression.
- (b) State a difference between multiplexers and decoders. Also state a use of each. **[2]**
- (c) Draw the truth table and a logic gate diagram for a **4 : 1** multiplexer and briefly explain its working. **[4]**

SECTION - B

Answer any *two* questions

Each program should be written in such a way that it clearly depicts the logic of the problem.

This can be achieved by using mnemonic names and comments in the program.

(Flowcharts and Algorithms are **not** required.)

The Programs must be written in Java.

Question 8

[10]

A class called **DigiNumber** has been defined to find the frequency of each digit present in a number and the sum of the digits. Some of the members of the class **DigiNumber** are given below:

Class name	:	DigiNumber
Data members/instance variables	:	
num	:	long integer for storing the number
Member functions :		
DigiNumber ()	:	constructor to assign 0 to num
DigiNumber (long n)	:	parameterised constructor to assign n to num
void digitfrequency()	:	find the frequency of each digit present in num and to display it
long sumDigits(long N)	:	to return the sum of the digits of the number stored in num using Recursive Technique
void printsum()	:	print sum of digits of num by invoking sumDigits(...)

Specify the class **DigiNumber**, giving details of the **two Constructor** and functions **void digitfrequency()**, **long sumDigits(long)** and **void printsum()**. Also write the **main()** function to create an object and call the member function accordingly.

Question 9

[10]

A class **Collection** contains an array of 100 integers. Using the following class description create an array with common elements from two integer arrays. Some of the members of the class are given below:

Class name	:	Collection
Data members/instance variables	:	
arr[]	:	an array of 100 integers.
len	:	length of the array
Member functions/methods	:	
Collection()	:	default constructor
Collection(int)	:	parameterised constructor to assign the length of array
void readArray()	:	to accept array elements.
Collection common(Collection)	:	return the Collection containing the common elements of the current Collection object and the Collection object passed as Parameter
void displayArray()	:	to display the array elements

Specify the class **Collection** giving the details of the constructors and the given. You **need not** write the main function.

Question 10

[10]

Write a Java program to input a sentence from the user in lowercase and removes the first and the last characters of every word in it.

Sample Input : i love java for school.
Sample Ouptut : ov av o choo

Some of the data members and member functions are given below:

Class name : Remove

Data members/instance variables:

sent : stores the sentence
rem : to store the new sentence
size : stores the length of the sentence

Member functions:

Remove() : default constructor
void readsentence() : to accept the sentence
void remfirstlast() : extract each word and remove the first and the last alphabet of the word and form a new sentence '**rem**' using the changed words
void display() : display the original sentence along with the new changed sentence.

Specify the class **Capitalize** giving details of the constructor **Remove ()**, **void readsentence()**, **void remfirstlast()** and **void display()**. Define the **main()** function to create an object and call the function accordingly to enable the task.

SECTION - C

Answer any *two* questions.

Each program/ Algorithm should be written in such a way that it clearly depicts the logic of the problem step wise. This can also be achieved by using pseudo codes.

(Flowcharts are **not** required.)

The Programs must be written in Java.

The Algorithm must be written in general/standard form, wherever required.

Question 11

[10]

A class **Salesman** defines the personal data of a salesman while another class **Sales** defines the bill number, name of the item, number of items sold and price of the item. The details of classes are as:

Class name	:	Salesman
Data members/instance variables	:	
name	:	string variable to store name of salesman
address	:	string variable to store address.
Member functions/methods	:	
Salesman()	:	constructor to assign null ("") to name and address.
void readDetails(String n, String ad)	:	to assign 'n' to name and 'ad' to address.
void show()	:	to print values of name and address using suitable headings
Class Name	:	Sales
Data members/instance variables	:	
billno, qty	:	integer to store bill number and quantity
price, psales	:	double type to store price and previous sales amount .
pname	:	string variable to store name of product
Member functions/methods	:	
void readDetails(int b, int q, double p, double s, String pr)	:	to assign 'b' to billno , 'q' to qty , 'p' to price , 's' to psales (previous sales) and 'pr' to pname
double calculate()	:	to calculate and return the total sales using the formula: p * qty + psales
void show()	:	to display values of pname, billno, qty, price, sales and total sales made by invoking suitable function.

Specify the class **Salesman** giving the details of the **constructor**, functions **void readDetails(...)** and **void show()**. Using the concept of **inheritance**, specify the class **Sales** giving the details of the functions **void readDetails(...)**, **double calculate()** and **void show()** function. You **do not** need to write the **main()** function.

Question 12

[10]

A circular queue is a linear data structure which enables the user to add integers from either ends, i.e. from front and rear, but allows removal only from the front end. Define a class **Cqueue** with the following details:

Class name : **Cqueue**

Data members/ instance variables :

que[] : array to hold up to 100 integer elements.
lim : stores the limit of the circular queue.
front : integer to point to the index of front end.
rear : integer to point to the index of the rear end.

Member functions/methods :

Cqueue(int l) : constructor to initialize the data members lim=l;
front = rear = 0

void insert(int val) : to add integers to the circular queue from both the ends
if possible else display a message ("Overflow")

int delete() : returns element from front if possible otherwise returns
-9999

Specify the class **Cqueue** giving details of the **constructor(int)** and the functions **void insert(int)**, **int remove()**. **The main() function and algorithm need not be written.**

Question 13

(a) A linked list is formed from the objects of the class,

[4]

```
class Node {
    int item;
    Node ptr;
}
```

Write an **Algorithm** Or a **Method** to find the sum of the values of all the elements in the linked list.

The method declaration is given below:

int sum(Node ptr_start)

(c) What is the worst case complexity of the following code segment:

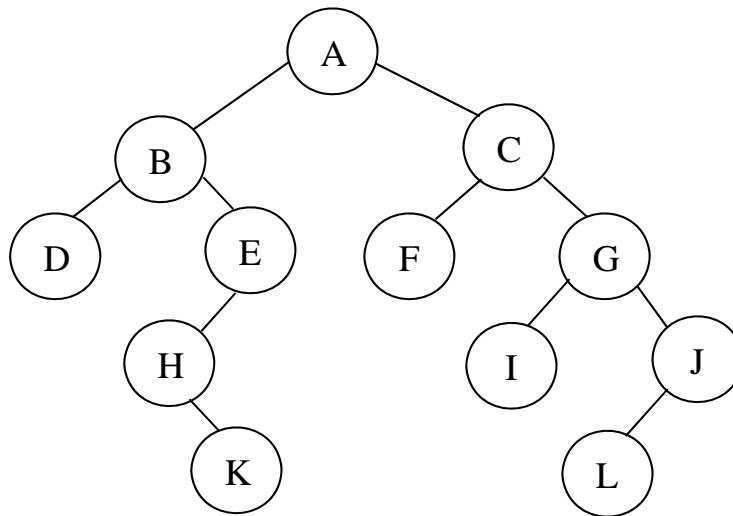
[2]

```
a = a + 1
for (i = 1 ; i <= N ; i++) {
    m = m + 2;
}

for (i = 1 ; i <= M ; i++) {
    for (i = 1 ; i <= N ; i++)
    {
        k = k + 1;
    }
}
```


(b) Answer the questions below for the given binary tree:

[4]



- (i) List the internal nodes of the Binary tree
- (ii) List the Leaf nodes of the Binary tree
- (iii) Draw right sub-tree of Node C
- (iv) Write Preorder traversal of the Binary Tree