COMPUTER SCIENCE
PAPER 1
(THEORY)

(Maximum Marks: 70)

(Time allowed: 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 **six** questions from Part-II, choosing **two** 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 (20 Marks)

Answer **all** *questions.*

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

Question 1

(a) Name and draw the logic gate represented by the following truth table, where A and B are inputs and X is the output. **[1]**

A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

(b) Write the canonical POS expression of: $F(P, Q) = \Pi (0,2)$ **[1]**

(c) Find the dual of: $X.Y + X.Y' = X + 0$ **[1]**

(d) If $F(A, B, C) = A'.B'.C' + A'.B.C'$ then find F' using De Morgan's Law. **[1]**

(e) If A= "It is cloudy" and B= "It is raining", then write the proposition for: **[1]**

(i) Contrapositive

(ii) Converse

This Paper consists of 10 printed pages.

Question 2

(a) What is an *Interface*? How is it different from a *class*? **[2]**

(b) A matrix ARR[– 4…6, 3….8] is stored in the memory with each element requiring 4 bytes of storage. If the base address is 1430, find the address of ARR[3][6] when the matrix is stored in **Row Major Wise**. **[2]**

(c) Convert the following *infix notation* to *postfix* form: **[2]**

(A + B * C) – (E * F / H) + J

(d) Compare the two complexities $O(n^2)$ and $O(2^n)$ and state which is better and why. **[2]**

(e) State the difference between *internal nodes* and *external nodes* of a binary tree structure. **[2]**

Question 3

The following function **Mystery( )** is a part of some class. What will the function Mystery( ) return when the value of **num=43629**, **x=3** and **y=4** respectively? Show the dry run/ working. **[5]**

```
int  Mystery( int num, int x, int y)
 {
      if(num<10)
          return num;
      else
        {
            int z = num % 10;
            if( z % 2 = = 0 )
                return z*x + Mystery( num/10,x,y);
            else
                return z*y + Mystery( num/10,x,y);
        }

   }
```

PART – II (50 Marks)

*Answer **six** questions in this part, choosing **two** questions from Section A, **two** from Section B and **two** from Section C.*

SECTION - A

*Answer **any two** questions.*

Question 4

(a) Given the Boolean function **F(A, B, C, D) = Σ (0, 2, 3, 4, 5, 8, 10, 11, 12, 13).**

(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). **[4]**

(ii) Draw the logic gate diagram for the reduced expression using only NAND gates. Assume that the variables and their complements are available as inputs. **[1]**

(b) Given the Boolean function: **F(P, Q, R, S) = π (0, 1, 2, 8, 9, 11, 13, 15).**

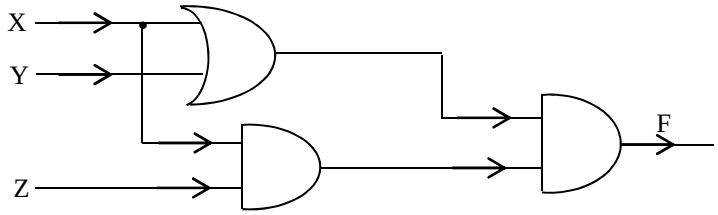
(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). **[4]**

(ii) Draw the logic gate diagram for the reduced expression using only NOR gates. Assume that the variables and their complements are available as inputs. **[1]**

Question 5

(a) How is a *decoder* different from a *multiplexer*? Write the truth table and draw the logic circuit diagram for a 3 to 8 decoder and explain its working. **[5]**

(b) From the logic circuit diagram given below, derive the Boolean expression and simplify it to show that it represents a logic gate. Name and draw the logic gate. **[3]**



(c) Using a truth table, state whether the following proposition is a Tautology, Contradiction or Contingency: **[2]**

~(P =>Q) <=> (~P ∨ Q)

Question 6

(a) The owner of a company pays bonus to his salesmen as per the criteria given below: **[5]**

- If the salesman works overtime for more than 4 hours but does not work on off days/holidays.

OR

- If the salesman works when festival sales are on and updates showroom arrangements.

OR

- If the salesman works on an off day/holiday when the festival sales are on.

The inputs are:

INPUTS	
O	Works overtime for more than 4 hours
F	Festival sales are on
H	Working on an off day/holiday
U	Updates showroom arrangements

(In all the above cases 1 indicates yes and 0 indicates no.)

Output : **X** [1 indicates yes, 0 indicates no for all cases]

Draw the truth table for the inputs and outputs given above and write the **POS** expression for **X(O,F,H,U)**.

(b) What is a *half adder*? Write the truth table and derive an SOP expression for *sum* and *carry* for a half adder. **[3]**

(c) Simplify the following expression, using Boolean laws: **[2]**

$$(X + Z) . (X.Y + Y.Z') + X.Z + Y$$

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 7

Design a class **ArmNum** to check if a given number is an Armstrong number or not. **[10]**

[A number is said to be Armstrong if sum of its digits raised to the power of length of the number is equal to the number]

Example : 371 $= 3^3 + 7^3 + 1^3$

1634 $= 1^4 + 6^4 + 3^4 + 4^4$

54748 $= 5^5 + 4^5 + 7^5 + 4^5 + 8^5$

Thus 371, 1634 and 54748 are all examples of Armstrong numbers.

Some of the members of the class are given below:

Class name	:	**ArmNum**
Data members/instance variables:		
n	:	to store the number
l	:	to store the length of the number
Methods/Member functions:		
ArmNum (int nn)	:	parameterized constructor to initialize the data member n=nn
int sum_pow(int i)	:	returns the sum of each digit raised to the power of the length of the number using **recursive technique** eg. 34 will return $3^2 + 4^2$ (as the length of the number is 2)
void isArmstrong( )	:	checks whether the given number is an Armstrong number by invoking the function *sum_pow( )* and displays the result with an appropriate message

Specify the class **ArmNum** giving details of the **constructor( )**, **int sum_pow(int)** and **void isArmstrong( ).** Define a **main( )** function to create an object and call the functions accordingly to enable the task.

Question 8

Design a class **MatRev** to reverse each element of a matrix. **[10]**

Example:

72	371	5
12	6	426
5	123	94

becomes

27	173	5
21	6	624
5	321	49

Some of the members of the class are given below:

Class name : **MatRev**

Data members/instance variables:

arr[ ][ ]	:	to store integer elements
m	:	to store the number of rows
n	:	to store the number of columns

Member functions/methods:

MatRev(int mm, int nn)	:	parameterised constructor to initialise the data members m = mm and n = nn
void fillarray( )	:	to enter elements in the array
int reverse(int x)	:	returns the reverse of the number x
void revMat(MatRev P)	:	reverses each element of the array of the parameterized object and stores it in the array of the current object
void show( )	:	displays the array elements in matrix form

Define the class **MatRev** giving details of the **constructor( ), void fillarray( ), int reverse(int)**, **void revMat(MatRev)** and **void show( )**. Define the **main( )** function to create objects and call the functions accordingly to enable the task.

Question 9

A class **Rearrange** has been defined to modify a word by bringing all the vowels in the word at the beginning followed by the consonants. **[10]**

Example: ORIGINAL becomes OIIARGNL

Some of the members of the class are given below:

Class name	:	**Rearrange**
Data member/instance variable:		
wrd	:	to store a word
newwrd	:	to store the rearranged word
Member functions/methods:		
Rearrange( )	:	default constructor
void readword( )	:	to accept the word in UPPER case
void freq_vow_con( )	:	finds the frequency of vowels and consonants in the word and displays them with an appropriate message
void arrange( )	:	rearranges the word by bringing the vowels at the beginning followed by consonants
void display( )	:	displays the original word along with the rearranged word

Specify the class **Rearrange**, giving the details of the **constructor( )**, **void readword( )**, **void freq_vow_con( ), void arrange( )** and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

SECTION – C

*Answer **any two** questions.*

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

This can be achieved by using comments in the program and mnemonic names or pseudo codes for algorithms. The programs must be written in Java and the algorithms must be written in general / standard form, wherever required / specified.

(Flowcharts are **not** required.)

Question 10

A super class **Record** contains names and marks of the students in two different single dimensional arrays. Define a sub class **Highest** to display the names of the students obtaining the highest mark. **[5]**

The details of the members of both the classes are given below:

Class name	:	**Record**
Data member/instance variable:		
n[ ]	:	array to store names
m[ ]	:	array to store marks
size	:	to store the number of students
Member functions/methods:		
Record(int cap)	:	parameterized constructor to initialize the data member size = cap
void readarray()	:	to enter elements in both the arrays
void display( )	:	displays the array elements
Class name:		**Highest**
Data member/instance variable:		
ind	:	to store the index
Member functions/methods:		
Highest(…)	:	parameterized constructor to initialize the data members of both the classes
void find( )	:	finds the index of the student obtaining the highest mark and assign it to '**ind**'
void display( )	:	displays the array elements along with the names and marks of the students who have obtained the highest mark

*Assume that the super class **Record** has been defined*. Using the **concept of inheritance**, specify the class **Highest** giving the details of the **constructor(…)**,**void find( )** and **void display( )**.

The super class, main function and algorithm need NOT be written.

Question 11

A linear data structure enables the user to add address from rear end and remove address from front. Define a class **Diary** with the following details:

Class name	**:**	**Diary**
Data members / instance variables:		
Q[ ]	:	array to store the addresses
size	:	stores the maximum capacity of the array
start	:	to point the index of the front end
end	:	to point the index of the rear end
Member functions:		
Diary (int max)	:	constructor to initialize the data member size=max, start=0 and end=0
void pushadd(String n)	:	to add address in the diary from the rear end if possible, otherwise display the message " NO SPACE"
String popadd( )	:	removes and returns the address from the front end of the diary if any, else returns "**?????**"
void show( )	:	displays all the addresses in the diary

(a) Specify the class **Diary** giving details of the functions **void pushadd(String)** and **String popadd( ).** Assume that the other functions have been defined. **[4]**

The main function and algorithm need NOT be written.

(b) Name the entity used in the above data structure arrangement. **[1]**

Question 12

(a) A linked list is formed from the objects of the class **Node**. The class structure of the Node is given below: **[2]**

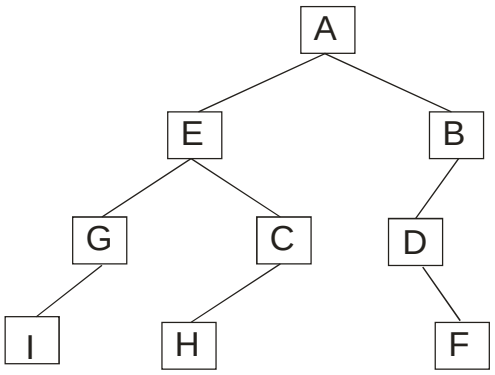
```
class   Node
{
        int num;
        Node next;
}
```

Write an *Algorithm* **OR** a *Method* to find and display the sum of even integers from an existing linked list.

The method declaration is as follows:

void SumEvenNode(Node str)

(b) Answer the following questions from the diagram of a Binary Tree given below:



(i) Write the pre-order traversal of the above tree structure. **[1]**

(ii) State the size of the tree. **[1]**

(iii) Name the siblings of the nodes E and G. **[1]**