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) State and verify distributive law using the truth table. **[2]**

(b) Why NAND gate is regarded as the universal gate? Draw the logic gate symbol and make the truth table for the two input NAND gate. **[2]**

(c) Find the complement of F (a, b, c, d) using De Morgan's laws. Show the relevant reasoning. **[2]**
F(a, b, c, d) = a + {(b + c).(b' + d')}.

(d) Simplify using law of Boolean algebra. At each step state clearly the laws used for simplifications. F = x.y + x.z + x.y.z **[2]**

(e) Given the Boolean function F(x, y, z) = $\sum(0, 2, 4, 5, 6)$. Reduce it using Karnaugh's map. **[2]**

Question 2

(a) What is a binary tree? **[1]**

(b) What do you understand by overflow? **[1]**

(c) What is an Interface? **[2]**

(d) Each element of an array **D[-15....20, 20....45]** requires 2 bytes of storage. If the array is stored in **column major form** and the base address of D[0][0] is 1200, Determine the location of D[1][40]. **[2]**

(e) Explain Worst Case Complexity of an algorithm. **[2]**

(f) Convert the following infix notation to its postfix form: **[2]**
A - B / (C ^ D) + (M * N)

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

Question 3

(a) The following functions **show()** and **calling()** are parts of some class. Assume that the parameter n is greater than **1** when the function is invoked. It returns value **1** when true otherwise returns **0**.
Show the dry run/working.

```
void calling()
{
 int f = 2;
 show(n, f);
}

int show(int n, int f)
{
 if(n == f)
       return 1;
 if(n % f ==0 || n==1)
       return 0;
 else
       return (show(n, f+1));
}
```

(i) What will the function **show()** returns when **calling()** is invoked with **n = 11**? **[2]**
(ii) What will the function **show()** returns when **calling()** is invoked with **n = 27**? **[2]**
(iii) In one line, state what the function **show()** is doing, apart from recursion. **[1]**

(b) The following is a function of some class. It returns 1 if the number is a perfect number otherwise it returns 0.

/* A perfect number is a number which is equal to the sum of its factors other than the number itself.*/

```
int PerfectNo (int n)
     {
       int?1?
       for(int j=1;?2?;j++)
         {
         if(?3?)
           sum?4?;
         }
       if(?5?)
         return1;
       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) = Σ (0, 1, 2, 5, 8, 9, 10)**

(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) = π (3, 4, 6, 7, 11, 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

For the selection in national level racing competition the selection committee has decided to select few candidates who satisfies at least **one** of the following conditions:

- The candidate is a female not below 18 years of age and has won prize at the state level.
- The candidate is a male of 18 years or above and has won the prize at the state level.
- The candidate is a male who is a member of racing organization and also National level player.
- The candidate is a female who has qualified in inter-school racing competition of a state.

THE INPUTS ARE:

A - The candidate is a male (1 indicates yes and 0 indicates no)

B - The candidate is 18 years and above (1 indicates yes and 0 indicates no)

C - The candidate belongs to racing organization or/and National level player (1 indicates yes and 0 indicates no)

D - The candidate is an inter-school state champion or state prize winner (1 indicates yes and 0 indicates no)

OUTPUT IS:

S : The candidate 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 **SOP** expression for S (A, B, C, D). **[5]**

(b) Reduce S (A, B, C, D) using Karnaugh's map. **[5]**

Draw the logic gate diagram for the reduced **SOP** expression for S (A, B, C, D).

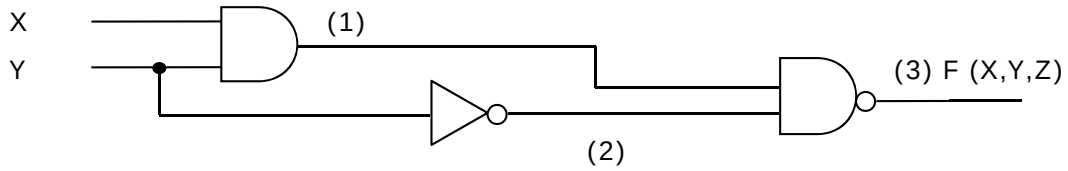
You may use gates with two or more inputs. Assume that variable and their complements are available as inputs.

Question 6

(a) Draw a logic diagram for the following function using only **NOR** gates: **[3]**
F(A, B, C) = (A' + B') . (B + C')

(b) Convert the following expression into canonical **Sum-Of-Product** form: **[3]**
F(X, Y, Z) = X'.Y + X'.Y' + Y.Z'

(c) From the logic circuit diagram given below, name the outputs (1), (2) and (3). Finally derive the Boolean expression and simplify it. **[4]**
Name and draw the logic gate.



Question 7

(a) Draw the truth table to prove the following proportional logic expression:
A <=> B = (A <=> B) ^ (B => A) **[3]**

(b) State a difference between a Tautology and Contradiction. **[2]**

(c) Draw the truth table and a logic gate diagram of Hexa-Decimal to Binary Encoder. **[5]**

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 **LCM** has been defined to find the LCM of two integers. Some of the members of the class **LCM** are given below:

Class name	**:**	**LCM**
Data members/instance variables	**:**	
n1, n2	:	integers whose LCM is to be found.
large, sm	:	integers to store the largest and the smallest from n1, n2
lcm	:	integer to store the LCM of the n1 and n2
Member functions :		
LCM ( )	:	constructor to assign initial values to the data members
void acceptData()	:	to accept the value of **n1** and **n2**
int getLCM()	:	to find the LCM of n1 and n2 using the **Recursive Technique** and store it in **lcm**. Return the value of **lcm**
void printData( )	:	to print the numbers **n1**, **n2** and the LCM

Specify the class **LCM**, giving details of the **two Constructor** and functions **void acceptData()**, **int getLCM()** and **void printData()**. Also write the **main( )** function to create an object and call the member function accordingly.

Question 9 **[10]**

The sum of angles is calculated as :

Let the first angle = 20 degrees 45 minutes
Let the second angle = 12 degrees 40 minutes

The sum of anlges will be 33 degrees 25 minutes. (where 60 minutes = 1 degree)

Design a class **Angle** with the following details:

Class name	**:**	**Angle**
Data members/instance variables	**:**	
deg	:	integer to store degrees.
min	:	integer to store minutes.
Member functions/methods	**:**	
Angle( )	:	constructor to assign 0 to **deg** and **min**.
void inputAngle()	:	to input values of **deg** and **min**.
void dispAngle()	:	to print the values of **deg** and **min** with proper message.
Angle sumAngle(Angle A, Angle B)	:	to find and return the sum of angles from objects **A** and **B** by using the above technique of adding angles

Specify the class **Angle** giving the details of the constructors and all the functions. You **need not** write the main function.

Question 10 **[10]**

Write a Java program to input a sentence from the user in lowercase and capitalizes the first character of every word present in it.

Sample Input **:** i love java for school.
Sample Ouptut **:** I Love Java For School.

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

Class name	**:**	**Capitalize**
Data members/instance variables	**:**	
sent	:	stores the sentence
cap	:	to store the new sentence
size	:	stores the length of the sentence
Member functions:		
Capitalize()	:	default constructor
void inpSentence()	:	to input the sentence
void capChar()	:	capitalizes the first character of every word and form a new sentence '**cap**'
void display()	:	display the original sentence along with the new changed sentence.

Specify the class **Capitalize** giving details of the constructor **Capitalize()**, **void inpSentence()**, **void capChar()** 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 **Student** defines the personal data of a student while another class **Marks** defines to the roll number, name of subject and marks obtained by the student. The details of classes are as:

Class name : **Student**

Data members/instance variables :

name, sex : string variables to store name and gender male/female
age : integer variable to store age of the student.

Member functions/methods :

Student( ) : constructor to assign initial values to the data members.
void inputDetails() : to accept values for data members.
void show( ) : to print personal data of student.

Class Name : **Marks**

Data members/instance variables :

rollno, marks : integers to store roll number and marks.
subject : string variable to store name of subject

Member functions/methods :

void readDetails() : to accept values for data members.
int point() : to return the point obtained according to the following:

Marks	Point
>=90	1
70 - 89	2
50 - 69	3
40 - 49	4
< 40	5

void show( ) : to display name, sex, age, roll number, marks, subject and point of the student by invoking suitable function.

Specify the class **Student** giving the details of the **constructor**, functions **void inputDetails()** and **void show()**. Using the concept of **inheritance**, specify the class **Marks** giving the details of the functions **void readDetails()**, **int point()** and **void show( )** function. You **do not** need to write the **main()** function.

Question 12 **[10]**

Define a class **Repeat** which allows the user to add elements from one end (rear) and remove elements from the other end (front) only. The details of the class **Repeat** is given below:

Class name **: Repeat**

Data members/ instance variables **:**

st[ ]	: array to hold up to 100 integer elements.
cap	: stores the capacity of the array.
f	: integer to point to the index of front end.
r	: integer to point to the index of the rear end.

Member functions/methods **:**

Repeat(int nn)	: constructor to initialize the data members cap=nn; f = 0, r = 0, and to create the integer array.
void pushvalue(int v)	: to add integers from the rear index if possible else display a message ("Overflow")
int popvalue()	: to remove and return element from front if possible otherwise returns -9999
void disp()	: Displays the elements present in the list. If the array is empty then display a message ("Underflow")

(a) Specify the class **Repeat** giving details of the **constructor()** and functions **void pushvalue(int)**, **int popvalue()** and **void disp(). The main() function and algorithm need not be written**.

(b) What is the common name of the entity described above?

(c) On what principle does this entity work?

Question 13

(a) A linked list is formed from the objects of the class, **[4]**

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

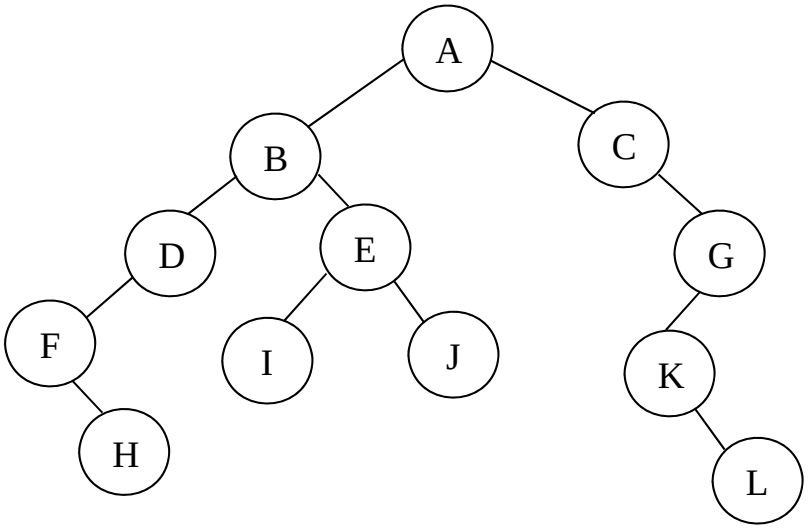
Write an *Algorithm* **OR** a *Method* to search for a value in a linked list.

The method declaration is given below:

void search(Node start, int value))

(b) Discuss with the help of an example why **$O(N^2)$** complexity is better than **$O(2^N)$**? **[2]**

(c) Answer the questions below for the given binary tree: **[4]**



(i) What is the level of this Binary tree

(ii) List the number of pairs of siblings of this Binary tree

(iii) Draw right sub-tree of Node C

(iv) Write Postorder traversal of the Binary Tree