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** questions from Section-B and **two** questions 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)   Using a truth table, verify the following expression: [2]

$$X + (Y + Z) = (X + Y) + Z$$

Also state the law.

(b)   Given, $F(X, Y, Z) = (X' + Y') . (Y + Z')$ [2]

write the function in canonical product-of-sum form.

(c)   Draw the truth table and logic circuit for a 2-input XNOR gate.   [2]

(d)   Find the complement of the following expression: [2]

$$X' + XY'$$

(e)   If $(X \Rightarrow Y)$ then write its :   [2]

(i)   Converse

(ii)   Contra positive

Question 2

(a) Differentiate between the keywords *extends* and *implements*. [2]

(b) State how a binary tree is a recursive data structure. [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** wise. [2]

(d) Convert the following infix notation to its postfix form: [2]

   A + ((B+C) + (D+E) * F) / G

(e) What is a constructor? State *one* difference between a constructor and any other member function of a class. [2]

Question 3

(a) The following function is a part of some class which computes and sorts an array arr[ ] in ascending order using the **bubble sort technique**. 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:

```
void    bubblesort(int arr[ ])
  {
        int  i, j, k, tmp;
        for(i= 0; ?1?; i++)
          {
                for(j = 0; ?2?; j++)
                  {
                    if(arr[j] > ?3?)
                      {
                        tmp       =    arr[j];
                        ?4?       =    arr[j+1];
                        arr[j+1]  =    ?5?;
                      }
                  }
          }
  }
```

(i) What is the expression or statement at **?1?** [1]

(ii) What is the expression or statement at **?2?** [1]

(iii) What is the expression or statement at **?3?** [1]

(iv) What is the expression or statement at **?4?** [1]

(v) What is the expression or statement at **?5?** [1]

2

(b)   The following function **witty( )** is a part of some class. What will be the   [5]
output of the function witty( ) when the value of n is "SCIENCE" and the value
of p is 5. Show the dry run / working:

```
void  witty(String n, int p)
{

    if (p < 0)
        System.out.println("");
    else
    {   System.out.println(n.charAt(p) + " . " );
        witty(n , p-1);
        System.out.print(n.charAt(p));
    }

}
```

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 (4, 6, 7, 10, 11, 12, 14, 15)**

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

  (ii) Draw the logic gate diagram of the reduced expression. Assume that the   [1]
variables and their complements are available as inputs.

(b) Given the Boolean function: **F(P,Q,R,S) = π (0, 5, 7, 8, 10, 12, 13, 14, 15)**

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

  (ii) Draw the logic gate diagram of the reduced expression. Assume that the   [1]
variables and their complements are available as inputs.

Question 5

The Principal of a school intends to select students for admission to class XI on the following criteria:

- Student is of the same school and has passed the class X Board Examination with more than 60% marks.

OR

- Student is of the same school, has passed the class X Board Examination with less than 60% marks but has taken active part in co-curricular activities.

OR

- Student is not from the same school but has either passed the class X Board Examination with more than 60% marks or has participated in sports at the National level.

The inputs are :

INPUTS	
S	Student of the same school.
P	Has passed the class X Board Examination with more than 60% marks.
C	Has taken active part in co-curricular activities.
T	Has participated in sports at the National level.

Output: **X** - Denotes admission status [1 indicates granted and 0 indicates refused in all the cases.]

(a) Draw the truth table for the inputs and outputs given above and write the **SOP** expression. [5]

(b) Reduce **X (S, P, C, T)** using Karnaugh's Map. [5]
Draw the logic gate diagram for the reduced **SOP** expression for X (S, P, C, T) using AND and OR gate. You may use gates with two or more inputs. Assume that the variable and their complements are available as inputs.

Question 6

(a) Verify algebraically if, [2]

$$X'Y'Z' + X'Y'Z + X'YZ + X'YZ' + XY'Z' + XY'Z = X'+Y'$$

(b) Represent the Boolean expression $X+YZ'$ with the help of NOR gates only. [2]

(c) Define the terms *Contingency, Contradiction* and *Tautology*. [3]

(d) Consider the following truth table where A and B are two inputs and X is the output:

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

(i) Name and draw the logic gate for the given truth table. [2]

(ii) Write the POS of X(A,B). [1]

Question 7

(a) Define Multiplexer and state *one* of its uses. Draw the logic diagram for a 4:1   **[4]**
Multiplexer.

(b) State how a *Half Adder* is different from a *Full Adder*. Also give their respective uses.   **[3]**

(c) Minimize the following expression using Boolean laws: **[3]**

$$Q . (Q' + P) . R . (Q + R)$$

Also draw the logic gate for the reduced expression.

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

A class Combine contains an array of integers which combines two arrays into a single   **[10]**
array including the duplicate elements, if any, and sorts the combined array. Some of the
members of the class are given below:

Class name	: **Combine**

Data members / instance variables:

com[ ]	:	integer array
size	:	size of the array

Member functions/methods:

Combine (int nn)	:	parameterized constructor to assign size = nn
void inputarray( )	:	to accept the array elements
void sort( )	:	sorts the elements of combined array in ascending order using the **selection sort** technique
void mix(Combine A, Combine B)	:	combines the parameterized object arrays and stores the result in the current object array along with duplicate elements, if any
void display( )	:	displays the array elements

Specify the class Combine giving details of the **constructor(int), void inputarray( ),
void sort( ), void mix(Combine, Combine)** and **void display( )**. Also define the **main( )**
function to create an object and call the methods accordingly to enable the task.

Question 9

Design a class VowelWord to accept a sentence and calculate the frequency of words that [10] begin with a vowel. The words in the input string are separated by a single blank space and terminated by a full stop. The description of the class is given below:

| Class name | : | VowelWord |

Data members / instance variables:

| str | : | to store a sentence |
| freq | : | store the frequency of the words beginning with a vowel |

Member functions:

VowelWord( )	:	constructor to initialize data members to legal initial value
void readstr( )	:	to accept a sentence
void freq_vowel( )	:	counts the frequency of the words that begin with a vowel
void display( )	:	to display the original string and the frequency of the words that begin with a vowel

Specify the class VowelWord giving details of the **constructor( ), void readstr( ), void freq_vowel( ) and void display( )**. Also define the **main( )** function to create an object and call the methods accordingly to enable the task.

Question 10

A happy number is a number in which the eventual sum of the square of the digits of the [10] number is equal to 1.

Example: $28 = (2)^2 + (8)^2 = 4 + 64 = 68$

   $68 = (6)^2 + (8)^2 = 36 + 64 = 100$

   $100 = (1)^2 + (0)^2 + (0)^2 = 1 + 0 + 0 = 1$

   Hence, 28 is a happy number.

Example: $12 = (1)^2 + (2)^2 = 1 + 4 = 5$

   Hence, 12 is not a happy number.

Design a class Happy to check if a given number is a happy number. Some of the members of the class are given below:

Class name : **Happy**

Data members/instance variables:

 : stores the number

Member functions:

Happy( ) : constructor to assign 0 to n

void getnum(int nn) : to assign the parameter value to the number n = nn

int sum_sq_digits(int x) : returns the sum of the square of the digits of the number x, using the **recursive technique**

void ishappy( ) : checks if the given number is a happy number by calling the function sum_sq_digits(int) and displays an appropriate message

Specify the class Happy giving details of the **constructor( )**, **void getnum(int)**, **int sum_sq_digits(int)** and **void ishappy( )**. Also define a **main( )** function to create an object and call the methods to check for happy number.

*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

Link is an entity which can hold a maximum of 100 integers. Link enables the user to add elements from the rear end and remove integers from the front end of the entity. Define a class Link with the following details:

Class name   :   Link

Data members/instant variables :

lnk[ ]	:	entity to hold the integer elements
max	:	stores the maximum capacity of the entity
begin	:	to point to the index of the front end
end	:	to point to the index of the rear end

Member functions:

Link(int mm)	:	constructor to initialize max = mm, begin = 0, end = 0
void addlink(int v)	:	to add an element from the rear index if possible otherwise display the message "OUT OF SIZE..."
int dellink( )	:	to remove and return an element from the front index , if possible otherwise display the message "EMPTY..." and return −99
void display( )	:	displays the elements of the entity

(a)   Specify the class Link giving details of the **constructor(int)**, **void addlink(int)**, **int dellink( )** and **void display( )**.   **[9]**

THE MAIN FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

(b)   What type of data structure is the above entity?   **[1]**

Question 12

A super class Detail has been defined to store the details of a customer. Define a **[10]** sub class Bill to compute the monthly telephone charge of the customer as per the chart given below:

NUMBER OF CALLS	RATE
1 – 100	Only rental charge
101 – 200	60 paisa per call + rental charge
201 – 300	80 paisa per call + rental charge
Above 300	1 rupee per call + rental charge

The details of both the classes are given below:

Class name : **Detail**

Data members / instance variables:

name	:	to store the name of the customer
address	:	to store the address of the customer
telno	:	to store the phone number of the customer
rent	:	to store the monthly rental charge

Member functions:

Detail(…),	:	parameterized constructor to assign values to data members
void show( )	:	to display the details of the customer

Class name : **Bill**

Data members /instance variables:

n	:	to store the number of calls
amt	:	to store the amount to be paid by the customer

Member functions :

Bill(…)	:	parameterized constructor to assign values to data members of both classes and to initialize amt = 0.0
void cal( )	:	calculates the monthly telephone charge as per the chart given above
void show( )	:	displays the details of the customer and amount to be paid

Specify the class Detail giving details of the **constructor( )** and **void show( )**. Using the **concept of inheritance,** specify the class Bill giving details of the **constructor( )**, **void cal( )** and **void show( )**.

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 p;
        String n;
        node  next;
    }
```
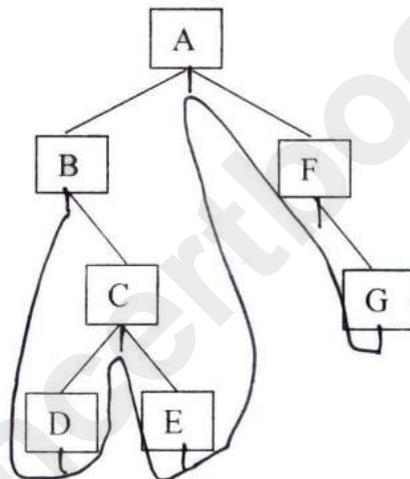
Write an *Algorithm* **OR** a *Method* to search for a name and display the contents of that node. The method declaration is given below:

void search(node start, String b)

(b) What is the role of constants in complexity? Explain briefly with an example.   [2]

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

(i)   External nodes of the tree.   [1]

(ii)   Parent of node D.   [1]

(iii) Inorder traversal of the tree.   [1]

(iv)   Right subtree of Node B.   [1]