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) State Associative law and prove it with the help of a truth table. **[1]**

(b) Draw the truth table to prove the proportional logic expression. **[1]**

$(X \Rightarrow Y) \wedge (Y \Rightarrow X) = X \Leftrightarrow Y$

(c) Find the dual for the Boolean equation: AB’ + BC’ + 1 = 1. **[1]**

(d) Convert the Boolean expression F(X,Y,Z) = X’Y’Z + X’YZ’ + XYZ into its cardinal form. **[1]**

(e) Minimize: F = XY + (XZ)’ +XY’Z using Boolean laws. **[1]**

Question 2

(a) Differentiate between *Stack data structure* and *Queue data structure*. **[2]**

(b) Convert the following infix notation to postfix notation **[2]**

A * (B / C) / E + F

(c) Define *Interface*. How is it different from a *Class*? **[2]**

(d) Each element of an array arr[15][20] requires 'W' bytes of storage. If the address of arr[6][8] is 4440 and the Base Address at arr[1][1] is 4000 , find the width 'W' of each cell in the array arr[ ][ ] when the array is stored as Column Major Wise. **[2]**

(e) Define Big 'O' notation. State the two factors which determine the complexity of an algorithm. **[2]**

Question 3

The following is a function of some class. What will be the output of the function **test ( )** when the value of count is equal to 4 ? Show the dry run / working. **[5]**

```
void test (int count)
   {
     if (count = = 0)
     System.out.println(" " );
     else
       {
         System.out.println( "Bye" + count);
         test( --count );
         System.out.println(" " + count);

        }
    }
```

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, 6, 8, 10, 11, 14, 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. Assume that the variables and their complements are available as inputs. **[1]**

(b) Given the Boolean function **F(P,Q,R,S) = π (5, 7, 8, 10, 12, 14, 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. Assume that the variables and their complements are available as inputs. **[1]**

Question 5

(a) Draw the logic diagram and truth table to encode the decimal numbers (**2, 3, 5, 7, 8**) and briefly explain its working **[5]**

(b) Simplify the following Boolean expression and draw the gate for the reduced expression: **[3]**

F= A'B + AB'C + A

(c) Define *Universal gates*. Give one example and show how it works as an OR gate. **[2]**

Question 6

(a) Draw a truth table with a 3 input combination which outputs 1 if there are odd number of 0's. Also derive an **SOP** expression for the output. Reduce the expression using Karnaugh Map. **[5]**

(b) Define *Proposition*. How does *tautology* differ from *contradiction*? **[3]**

(c) Draw the logic diagram of 4:1 Multiplexer. **[2]**

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

A class **Composite** contains a two dimensional array of order [m x n]. The maximum value possible for both 'm' and 'n' is 20. Design a class **Composite** to fill the array with the first (m x n) composite numbers in column wise. The details of the members of the class are given below: **[10]**

Class name	:	**Composite**
Data members/instance variables:		
arr[ ] [ ]	:	stores the composite numbers column wise
m	:	integer to store the number of rows
n	:	integer to store the number of columns
Member functions/methods:		
Composite(int mm, int nn)	:	to initialize the size of the matrix m=mm and n=nn
int isComposite(int p)	:	returns 1 if number is composite otherwise returns 0
void fill ( )	:	to fill the elements of the array with the first (m × n) composite numbers in column wise
void display( )	:	displays the array in a matrix form

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

Question 8

Design a class **Sort** which enables a word to be arranged in alphabetical order. The details of the members of the class are given below : **[10]**

Class name	:	**Sort**
Data members/instance variables:		
str	:	stores a word
len	:	to store the length of the word
Methods/Member functions:		
Sort( )	:	default constructor
void readword( )	:	to accept the word
void arrange ( )	:	to arrange the word in alphabetical order using any standard sorting technique.
void display( )	:	displays the original word along with the sorted word

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

Question 9

A **Special** number is a number in which the sum of the factorial of its digits is equal to the number. **[10]**

Example: 145 (1! + 4! + 5! = 145). Thus, 145 is a Special number.

Design a class **Special** to check if the given number is a Special number or not. Some of the members of the class are given below:

Class name	:	**Special**
Data member/instance variable:		
n	:	integer to store number
Member functions/methods:		
Special( )	:	default constructor
void read( )	:	to accept the number
int factorial(int x)	:	return the factorial of a number using **recursive technique**
boolean isSpecial( )	:	checks for the special number by invoking the function factorial( ) and returns true if Special, otherwise returns false
void display( )	:	displays the result with an appropriate message

Specify the class **Special**, giving details of the **Constructor**, **void read( ), int factorial(int), boolean isSpecial( ) and void display( )**. Define the **main()** function to create an object and call the member function according 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 step wise.
This can also be achieved by using comments in the program and mnemonic names or pseudo codes for algorithms. The program must be written in Java and the algorithms must be written in general / standard form, wherever required / specified.
(Flowcharts are **not** required.)

Question 10

An interface **Shape** is defined with a method **area()** which returns the area of the implementing shape. **[5]**

Create the classes **Circle** and **Rectangle** which implement the interface **Shape**. These classes have attributes which reflect their dimensions (radius for a circle, height and width for a rectangle) which are set by their constructors.

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

Interface name	:	**Shape**
Member functions/methods:		
double area()	:	returns the area of the implementing shape
Class name	:	**Circle**
Data members/instance variables:		
radius	:	to store radius of the circle in decimal
Member functions/methods:		
Circle(int r)	:	parameterized constructor to initiallize radius=r
double area()	:	to calculate area of the circle [area of a circle is 3.14*radius*radius]
Class name	:	**Rectangle**
Data members/instance variables:		
length	:	to store length of the rectangle in decimal
breadth	:	to store breadth of the rectangle in decimal
Member functions / methods		
Rectangle(int l, int b)	:	parameterized constructor to initialize length=l, breadth=b
double area()	:	to calculate area of the rectangle [area of a rectangle is length*breadth]

*Assume that the Interface **Shape** has been defined*. Using the **concept of inheritance**, specify the classes **Circle** and **Rectangle** giving details of their **constructor**s and **double area( )** respectively.

The interface, main function and algorithm need NOT be written.

Question 11

Circular Queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. **[5]**

Define a class **Cqueue** with the following details:

Class name	:	**Cqueue**
Data member/instance variable:		
ele[ ]	:	array to hold the integer elements
cap	:	stores the maximum capacity of the array
front	:	to point the index of the front
rear	:	to point the index of the rear.
Member functions/methods:		
Cqueue(int max)	:	constructor to initialize the data member cap = max, front = rear = 0 and create the integer array
void insert(int v)	:	to add integers from the front index if possible else display the message(“full from rear”)
int delete( )	:	to remove and return elements from rear, if any, else returns -999
void display()	:	to display elements of circular queue

Specify the class **Cqueue** giving the details of **void insert(int)** and **int delete( )**. Assume that the other functions have been defined.

The main( ) function and algorithm need NOT be written.

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]**

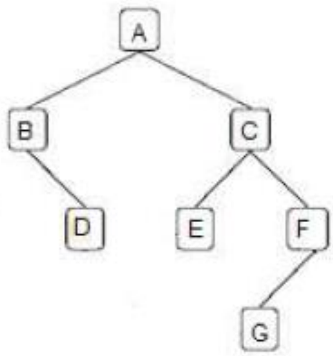
```
Class Node
{
    int num;
    Node next;
}
```

Write an *Algorithm* **OR** a *Method* to insert a node at the beginning of an existing linked list.

The method declaration is as follows:

void InsertNode(Node starPtr, int n)

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



(i) Name the Root and the leaves of the tree. **[1]**

(ii) Write the post order traversal of the tree. **[1]**

(iii) Separate the Internal nodes and the External nodes of the tree. **[1]**

(NOTE: The total weightage of questions in the Question Paper will be as indicated in the Specimen Paper. However, breakup of subparts in questions may vary from year to year.)