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 [ ].

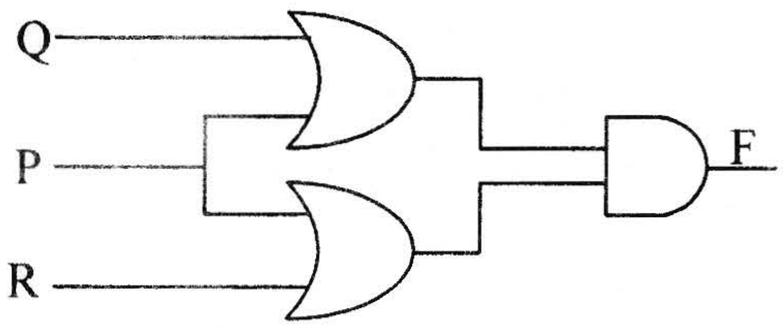
PART I

Answer **all** *questions.*

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

Question 1

(a) From the logic circuit diagram given below, find the output 'F' and simplify it. **[2]**
Also, state the law represented by the logic diagram.



(b) Write the truth table for a 2-input conjunction and disjunction in a proposition. **[2]**

(c) Find the complement of $XY'Z + XY + YZ'$ **[2]**

(d) Convert the following expression into its canonical POS form : **[2]**

$$F(A,B) = (A + B) \cdot A'$$

(e) Minimise the following Boolean expression using the Karnaugh map: **[2]**

$$F(A,B,C) = \overline{A}\,B\,\overline{C} + \overline{A}BC + AB\overline{C} + ABC$$

This Paper consists of 11 printed pages and 1 blank page.

Question 2

(a) State *two* advantages of using the *concept of inheritance* in Java. **[2]**

(b) An array AR [–4 .... 6, –2....12] , stores elements in **Row Major Wise**, with the address AR[2][3] as 4142. If each element requires 2 bytes of storage, find the Base address. **[2]**

(c) State the sequence of traversing a binary tree in: **[2]**

(i) preorder

(ii) postorder

(d) Convert the following infix expression into its postfix form: **[2]**

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

(e) State the difference between the functions *int nextInt( )* and *boolean hasNextInt( )*. **[2]**

Question 3

(a) The following functions are part of some class:

```
void fun1(char s[ ],int x)
  {  System.out.println(s);
     char temp;
     if(x<s.length/2)
        {  temp=s[x];
           s[x]=s[s.length-x-1];
           s[s.length-x-1]=temp;
           fun1(s,x+1);
        }
  }
  void fun2(String n)
  {  char c[ ]=new char[n.length( )];
     for(int i=0;i<c.length; i++)
       c[i]=n.charAt(i);
     fun1(c,0);
  }
```

(i) What will be the output of **fun1( )** when the value of s[ ]={'J','U','N','E'} and x=1? **[2]**

(ii) What will be the output of **fun2( )** when the value of n ="SCROLL" ? **[2]**

(iii) State in one line what does the function **fun1( )** do apart from recursion. **[1]**

(b) The following is a function of some class which sorts an integer array a[ ] in ascending order using *selection sort* technique. There are some places in the code marked by **?1?, ?2?, ?3?, ?4?, ?5?** which may be replaced by a statement/expression so that the function works properly:

```
void selectsort(int [ ]a)
  {
   int i,j,t,min,minpos;
   for( i=0;i<?1?;i++)
    {
        min=a[i];
        minpos = i;
       for(j=?2?;j<a.length;j++)
        {
         if(min>a[j])
          {
          ?3? = j;
          min = ?4?;
          }
       }
          t=a[minpos];
          a[minpos]=a[i];
          a[i]= ?5?;
       }

    for(int k=0;k<a.length;k++)
      System.out.println(a[k]);
  }
```

(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]**

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) = $\sum$ (0, 1, 2, 3, 5, 6, 7, 10, 13, 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 **P(A, B, C, D) = Π (0, 1, 2, 3, 5, 6, 7, 10, 13, 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 school intends to select candidates for the Inter-School Athletic Meet, as per the criteria given below:

- The candidate is from the Senior School and has participated in an Inter-School Athletic Meet earlier.

OR

- The candidate is not from the Senior School, but the height is between 5 ft. and 6 ft. and weight is between 50 kg. and 60 kg.

OR

- The candidate is from the senior school and has height between 5 ft. and 6 ft., but the weight is not between 50 kg. and 60 kg.

The inputs are:

INPUTS	
S	Student is from the Senior School
W	Weight is between 50 kg. and 60 kg.
H	Height is between 5 ft. and 6 ft.
A	Taken part in Inter-School Athletic Meet earlier

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

Output: **X** - Denotes the selection criteria [1 indicates selected and 0 indicates rejected in all cases.]

(a) Draw the truth table for the inputs and outputs given above and write the **SOP** expression for X(S, W, H, A). **[5]**

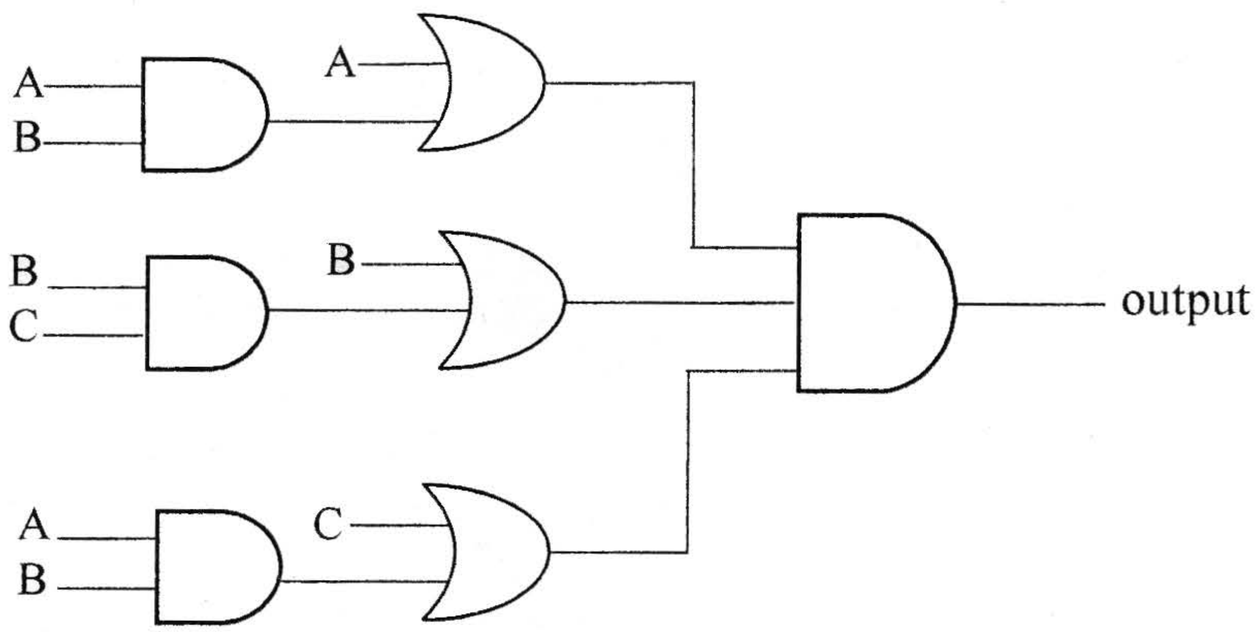
(b) Reduce **X(S, W, H, A)** using Karnaugh map. **[5]**

Draw the logic gate diagram for the reduced **SOP** expression for X(S, W, H, A) 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) With the help of a logic diagram and a truth table explain a *Decimal to Binary* encoder. **[4]**

(b) Derive a Boolean expression for the logic diagram given below and simplify it. **[3]**



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

$$F(A, B, C, D) = (A' + C)\,(A' + C')\,(A' + B + C'D)$$

Question 7

(a) Differentiate between XNOR and XOR gates. Draw the truth table and logic diagrams of 3 input XNOR gate. **[4]**

(b) Differentiate between a *proposition* and *wff*. **[2]**

(c) Define *Half Adder*. Construct the truth table and a logic diagram of a Half Adder. **[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

A class **Mixer** has been defined to merge two sorted integer arrays in ascending order. **[10]**
Some of the members of the class are given below:

Class name	:	**Mixer**
Data members/instance variables:		
int arr[ ]	:	to store the elements of an array
int n	:	to store the size of the array
Member functions:		
Mixer(int nn)	:	constructor to assign n=nn
void accept( )	:	to accept the elements of the array in ascending order without any duplicates
Mixer mix(Mixer A)	:	to merge the current object array elements with the parameterized array elements and return the resultant object
void display( )	:	to display the elements of the array

Specify the class **Mixer,** giving details of the **constructor(int)**, **void accept( )**, **Mixer mix(Mixer)** and **void display( )** . Define the **main( )** function to create an object and call the function accordingly to enable the task.

Question 9

A class **SeriesSum** is designed to calculate the sum of the following series:

$$\text{Sum} = \frac{x^2}{1!} + \frac{x^4}{3!} + \frac{x^6}{5!} + ... \frac{x^n}{(n-1)!}$$

Some of the members of the class are given below:

Class name : **SeriesSum**

Data members/instance variables:

x	:	to store an integer number
n	:	to store number of terms
sum	:	double variable to store the sum of the series

Member functions:

SeriesSum(int xx, int nn)	:	constructor to assign x=xx and n=nn
double findfact(int m)		to return the factorial of m using **recursive technique.**
double findpower(int x, int y)	:	to return x raised to the power of y using **recursive technique**.
void calculate( )	:	to calculate the sum of the series by invoking the recursive functions respectively
void display( )	:	to display the sum of the series

(a) Specify the class **SeriesSum,** giving details of the **constructor(int, int), double findfact(int), double findpower(int, int), void calculate( )** and **void display( ).** Define the **main( )** function to create an object and call the functions accordingly to enable the task. **[8]**

(b) State the *two* differences between *iteration* and *recursion*. **[2]**

Question 10

A sequence of fibonacci strings is generated as follows: **[10]**

S_0= "a", S_1= "b", $S_n = S_{(n-1)} + S_{(n-2)}$ where '+' denotes concatenation. Thus the sequence is:
a, b, ba, bab, babba, babbabab, ... n terms.

Design a class **FiboString** to generate fibonacci strings. Some of the members of the class are given below:

Class name	:	**FiboString**
Data members/instance variables:		
x	:	to store the first string
y	:	to store the second string
z	:	to store the concatenation of the previous two strings
n	:	to store the number of terms
Member functions/methods:		
FiboString( )	:	constructor to assign x="a", y="b" and z="ba"
void accept( )	:	to accept the number of terms 'n'
void generate( )	:	to generate and print the fibonacci strings. The sum of ('+' ie concatenation) first two strings is the third string. Eg. "a" is first string, "b" is second string then the third will be "ba", and fourth will be "bab" and so on.

Specify the class **FiboString,** giving details of the **constructor( ), void accept( )** and **void generate( )**. 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 11

A super class **Stock** has been defined to store the details of the stock of a retail store. Define a subclass **Purchase** to store the details of the items purchased with the new rate and updates the stock. Some of the members of the classes are given below: **[10]**

Class name : **Stock**

Data members/instance variables:

item	:	to store the name of the item
qty	:	to store the quantity of an item in stock
rate	:	to store the unit price of an item
amt	:	to store the net value of the item in stock

Member functions:

Stock (…)	:	parameterized constructor to assign values to the data members
void display( )	:	to display the stock details

Class name : **Purchase**

Data members/instance variables:

pqty	:	to store the purchased quantity
prate	:	to store the unit price of the purchased item

Member functions / methods

Purchase(…)	:	parameterized constructor to assign values to the data members of both classes
void update ( )	:	to update stock by adding the previous quantity by the purchased quantity and replace the rate of the item if there is a difference in the purchase rate. Also update the current stock value as: (quantity * unit price)
void display( )		to display the stock details before and after updation

Specify the class **Stock**, giving details of the **constructor()** and **void display( )**. Using **concept of inheritance**, specify the class **Purchase,** giving details of the **constructor(), void update( )** and **void display( )**.

The main function and algorithm need not be written.

Question 12

A stack is a linear data structure which enables the user to add and remove integers from one end only, using the concept of LIFO(Last In First Out). An array containing the marks of 50 students in ascending order is to be pushed into the stack. **[10]**

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

Class name	:	**Array_to_Stack**
Data members/instance variables:		
m[ ]	:	to store the marks
st[ ]	:	to store the stack elements
cap	:	maximum capacity of the array and stack
top	:	to point the index of the topmost element of the stack
Methods/Member functions:		
Array_to_Stack (int n)	:	parameterized constructor to initialize cap = n and top = -1
void input_marks( )	:	to input the marks from user and store it in the array m[ ] in ascending order and simultaneously push the marks into the stack st[ ] by invoking the function pushmarks( )
void pushmarks(int v)	:	to push the marks into the stack at top location if possible, otherwise, display "not possible"
int popmarks( )	:	to return marks from the stack if possible, otherwise, return -999
void display( )	:	To display the stack elements

Specify the class **Array_to_Stack**, giving the details of the **constructor(int)**, **void input_marks( )**, **void pushmarks(int)**, **int popmarks( )** and **void display( ).**

The main function and the algorithm need not be written.

Question 13

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

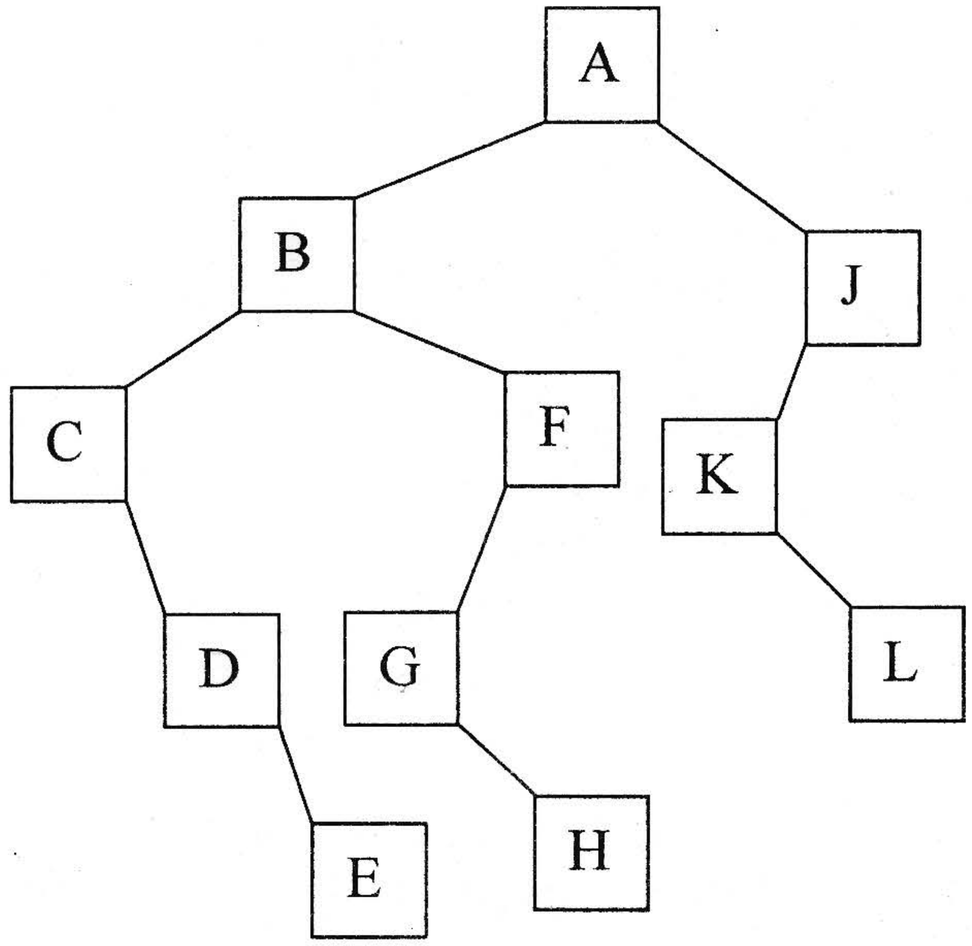
```
class Node
{
        int     number;
        Node    nextNode;
}
```

Write an *Algorithm* **OR** a *Method* to add a node at the end of an existing linked list. The method declaration is as follows:

void addnode (Node start, int num)

(b) Define the terms *complexity* and *big 'O' notation.* **[2]**

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



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

(ii) Left subtree **[1]**

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

(iv) Size of the tree. **[1]**