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

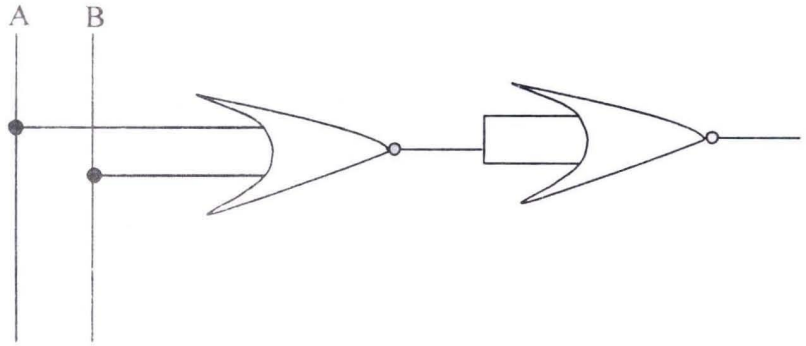
(a) State the two Absorption laws. Verify any one of them using truth table. **[2]**

(b) Reduce the following expression: **[2]**

$F(A,B,C) = \sum (0,1,2,3,4,5,6,7)$

Also find the complement of the reduced expression.

(c) Name the logic gate for the following circuit diagram and write its truth table. **[2]**



(d) Using truth table, verify whether the following is true or false: **[2]**

$(p \Rightarrow q) = (\bar{q} \Rightarrow \bar{p})$

(e) If A=1, B=0, C=1 and D=1 find its : **[2]**

(i) Maxterm

(ii) Minterm

Question 2

(a) How can we override a method in inheritance? [2]

(b) A square matrix A[m x m] is stored in the memory with each element requiring 2 bytes of storage. If the base address A[1][1] is 1098 and the address at A [4][5] is 1144, determine the order of the matrix A[m x m] when the matrix is stored **Column Major** wise. [2]

(c) What is Big O notation? [2]

(d) What is an exception? [2]

(e) Convert the following infix expression to its postfix form: [2]

a + b * c – d / e

Question 3

(a) The following is a part of some class. What will be the output of the function **mymethod( )** when the value of counter is equal to 3? Show the dry run / working. [5]

```
void mymethod(int counter)
 {
  if ( counter = = 0 )
       System.out.println(" ");
  else
   {
     System.out.println("Hello"+counter);
     mymethod(– –counter ) ;
     System.out.println("  "+counter);
   }
 }
```

(b) The following function is a part of some class which computes and returns the greatest common divisor of any two numbers. There are some places in the code marked by **?1?, ?2?, ?3?, ?4?** and **?5?** which must be replaced by statement/expression so that the function works correctly.

```
int gcd(int a, int b)
 {
      int r;
      while(?1?)
       {
          r=?2?;
          b=?3?;
          a=?4?;
       }
      if(a==0)
          return ?5?;
      else
          return -1;
 }
```

(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) State the principle of Duality. Give the dual of the following : [3]

$(A'.B)+(C.1)=(A'+C).(B+C)$

(b) Reduce the Boolean expressions to their simplest forms: [4]

(i) $\{(C.D)'+ A\} + A + C.D + A.B$

(ii) $A.\{B+C(A.B+A.C)'\}$

(c) Verify using a truth table if : [3]

$(A \odot B \odot C)' = A \oplus B \oplus C$

Question 5

(a) Given $\mathbf{F(P, Q, R, S) = \pi(2, 3, 6, 7, 9, 11, 12, 13, 14, 15)}$ [5]

Reduce the above expression by using ***four*** variable Karnaugh's Map. Draw the logic gate diagram of the reduced expression using NOR gate only.

(b) Given $\mathbf{F(A, B, C, D) = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} + \bar{A}\,\bar{B}\,\bar{C}\,D + A\,\bar{B}\,\bar{C}\,\bar{D} + A\,\bar{B}\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,\bar{D} + \bar{A}\,B\,C\,\bar{D}}$. [5]

Reduce the above expression by using ***four*** variable Karnaugh's Map. Draw the logic gate diagram of the reduced expression using NAND gate only.

Question 6

(a) Show with the help of a logic diagram how a NAND gate is equivalent to an OR gate. [3]

(b) Verify if the following is valid : [3]

$(a \Rightarrow b) \wedge (a \Rightarrow c) = a \Rightarrow (b \wedge c)$

(c) What is a Decoder? Draw the truth table and logic circuit diagram for a 2 to 4 Decoder. [4]

Question 7

(a) What is a Full Adder? Draw the truth table for a Full adder. Also derive SOP expression for the Full Adder and draw its logic circuit. **[4]**

(b) State how a Decoder is different from a Multiplexer. Also state one use of each. **[3]**

(c) Convert the following cardinal expression into its canonical form and reduce it using Boolean laws: **[3]**

$$F(L,M,O,P) = \pi\,(0,2,8,10)$$

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

Input a sentence from the user and count the number of times, the words "an" and "and" are present in the sentence. Design a class **Frequency** using the description given below: **[10]**

Class name	:	**Frequency**
Data Members / variables:		
text	:	stores the sentence
countand	:	to store the frequency of the word "and"
countan	:	to store the frequency of the word "an"
len	:	stores the length of the string
Member functions/methods:		
Frequency( )	:	constructor to initialize the instance variables
void accept(String n)	:	to assign n to text, where the value of the parameter n should be in lower case.
void checkandfreq( )	:	to count the frequency of "and"
void checkanfreq( )	:	to count the frequency of "an"
void display( )	:	to display the number of "and" and "an" with appropriate messages.

Specify the class **Frequency** giving details of the **constructor( ), void accept(String), void checkandfreq( ), void checkanfreq( )** and **void display( )**. Also define the **main( )** function to create an object and call methods accordingly to enable the task.

Question 9

A class **DeciOct** has been defined to convert a decimal number into its equivalent octal number. Some of the members of the class are given below:

Class name	:	**DeciOct**
Data Members / instance variables:		
n	:	stores the decimal number
oct	:	stores the octal equivalent number
Member functions:		
DeciOct( )	:	constructor to initialize the data members n=0, oct = 0.
void getnum(int nn)	:	assign nn to n
void deci_oct( )	:	calculates the octal equivalent of 'n' and stores it in oct using the **recursive technique**
void show( )	:	displays the decimal number 'n', calls the function deci_oct( ) and displays its octal equivalent.

(a) Specify the class **DeciOct**, giving details of the **constructor( ), void getnum(int), void deci_oct( )** and **void show( ).** Also define a **main( )** function to create an object and call the functions accordingly to enable the task. **[8]**

(b) State any two disadvantages of using recursion. **[2]**

Question 10

You are given a sequence of N integers, which are called as pseudo arithmetic sequences (sequences that are in arithmetic progression). **[10]**

Sequence of N integers : 2, 5, 6, 8, 9, 12

We observe that 2 + 12 = 5 + 9 = 6 + 8 = 14.

The sum of the above sequence can be calculated as 14 x 3 = 42.

For sequence containing an odd number of elements the rule is to double the middle element, for example 2, 5, 7, 9, 12

= 2 + 12 = 5 + 9 = 7 + 7 = 14.

14 × 3 = 42 [middle element = 7]

A class **Pseudoarithmetic** determines whether a given sequence is a pseudo-arithmetic sequence.

The details of the class are given below:

Class name	:	**Pseudoarithmetic**
Data Members / instance variables:		
n	:	to store the size of the sequence
a [ ]	:	integer array to store the sequence of numbers
ans, flag	:	store the status
sum	:	store the sum of sequence of numbers
r	:	store the sum of the two numbers
Member functions:		
Pseudoarithmetic( )	:	default constructor
void accept(int nn)	:	to assign nn to n and to create an integer array. Fill in the elements of the array.
boolean check( )	:	return true if the sequence is a pseudo-arithmetic sequence otherwise returns false

Specify the class **Pseudoarithmetic**, giving the details of the **constructor( )**, **void accept(int)** and **boolean check( ).** Also define a **main( )** function to create an object and call the member functions 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.

Question 11

A super class **Record** has been defined to store the names and ranks of 50 students. Define a sub-class **Rank** to find the highest rank along with the name. The details of both classes are given below: **[10]**

Class name	:	**Record**
Data Members / instance variables:		
name[ ]	:	to store the names of students
rnk[ ]	:	to store the ranks of students
Member functions:		
Record( )	:	constructor to initialize data members
void readvalues( )	:	to store the names and ranks
void display( )	:	displays the names and the corresponding ranks
Class name	:	**Rank**
Data members /instance variables:		
index	:	integer to store the index of the topmost rank
Member functions :		
Rank( )		constructor to invoke the base class constructor and to initialize index = 0
void highest( )		finds the index/location of the topmost rank and stores it in index without sorting the array.
void display( )		displays the names and ranks along with the name having the topmost rank.

Specify the class **Record** giving details of the **constructor( ), void readvalues( )** and **void display( )**. Using the **concept of inheritance**, specify the class **Rank** giving details of **constructor( ), void highest( )** and **void display( )**.

THE MAIN( ) FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

Question 12

Stack is a kind of data structure which can store elements with the restriction that an element can be added or removed from the top only.

The details of the class **Stack** is given below:

Class name	:	**Stack**
Data Members / instance variables:		
st[ ]	:	the array to hold the names
size	:	the maximum capacity of the string array
top	:	the index of the topmost element of the stack
ctr	:	to count the number of elements of the stack
Member functions:		
Stack( )	:	default constructor
Stack(int cap)	:	constructor to initialize size = cap and top = –1
void pushname(String n)	:	to push a name into the stack. If the stack is full, display the message "OVERFLOW"
String popname( )	:	removes a name from the top of the stack and returns it. If the stack is empty, display the message "UNDERFLOW"
void display( )	:	Display the elements of the stack.

(a) Specify class **Stack** giving details of the **constructors( ), void pushname (String n), String popname( )** and **void display( ).** **[8]**

THE MAIN( ) FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

(b) Under what Principle does the above entity work? **[2]**

Question 13

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

```
class Node
 {
    int info ;
    Node link ;
 }
```

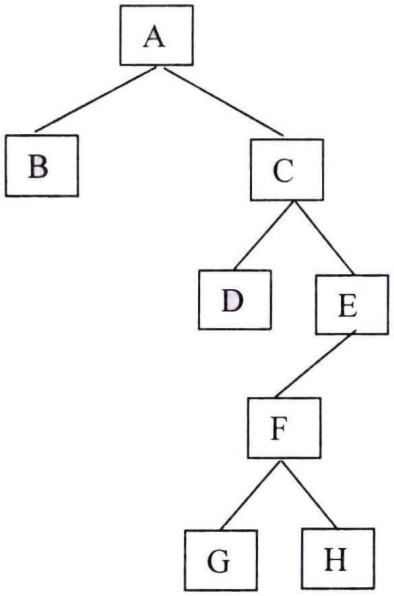
Write an algorithm **OR** a Method for deleting a node from a linked list.

The method declaration is given below :

void deletenode (Node start)

(b) Distinguish between worst-case and best case complexity of an algorithm. **[2]**

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



(i) Write the Postorder tree traversal **[1]**

(ii) Name the Leaves of the tree **[1]**

(iii) Height of the tree **[1]**

(iv) Root of the tree **[1]**