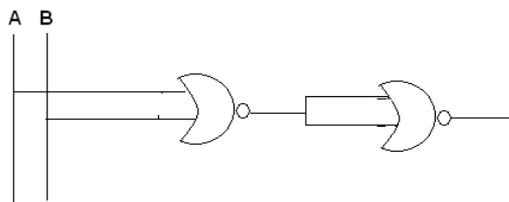


**ISC 2011  
COMPUTER SCIENCE PAPER 1  
THEORY**

**Question 1.**

- a) State the two absorption laws. Verify any one of them using truth table.
- b) Reduce the following expression :  
 $F(A,B,C) = \sum(0,1,2,3,4,5,6,7)$   
 Also find the complement of the reduced expression.
- c) Name the logic gate for the following circuit diagram and write its truth table.



- d) Using truth table, verify whether the following is true or false:  
 $(p \rightarrow q) = (q' \rightarrow p')$
- e) If A=1, B=0, C=1 and D=1 find its
  - i) maxterm
  - ii) minterm

**[2 x 5 =10]**

**Question 2.**

- a) How can we override a method in inheritance?
- 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[mxm] when the matrix is stored column majorwise.
- c) What is Big O Notation?
- d) What is an exception?
- e) Convert the infix expression to its postfix form:  
 $A+B*C-D/E$

**[ 2 x 5 =10]**

**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.

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

**[ 5]**

- 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?  
 ii) What is the expression or statement at ?2?  
 iii) What is the expression or statement at ?3?  
 iv) What is the expression or statement at ?4?  
 v) What is the expression or statement at ?5?

[ 1 x 5 = 5]

### Part II

Answer seven questions in this Part, choosing three questions from Section A and two questions from Section B and two questions from Section C.

### Section A

Answer any three questions from this Section

#### Question 4.

- a) State the principle of duality. Give the dual of the following:  
 $(A'B) + (C.1) = (A'+C)(B+C)$  [ 3 ]
- b) Reduce the following Boolean expressions to their simplest forms:-  
 i)  $\{(CD)' + A\} + A + C.D + A.B$   
 ii)  $A.\{B+C(A.B + A.C)'\}$  [ 4 ]
- c) Verify using a truth table if:  
 $(A \oplus B \oplus C)' = A \oplus B \oplus C$  [ 3 ]

**Question 5.**

- a) Given  $F(P,Q,R,S) = \pi(2,3,6,7,9,11,12,13,14,15)$   
Reduce the above expression using four variable K-map. Draw the logic gate diagram of the reduced expression using NOR gates only. [ 5 ]
- b) Given  $F(A,B,C,D) = A'B'C'D' + A'B'C'D + AB'C'D' + AB'C'D + A'BC'D' + A'BCD'$ .  
Reduce the above expression by using four variable K-map. Draw the logic gate diagram of the reduced expression using NAND gates only. [ 5 ]

**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:  
 $(A \rightarrow B) \wedge (A \rightarrow C) = A \rightarrow (B \wedge C)$  [ 3 ]
- 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:  $F(L,M,O,P) = \pi(0,2,8,10)$  [ 3 ]

**Section B**

Answer any 2 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.

**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:

**Class name** : Frequency

**Data Members**

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

Frequency()	: constructor to initialize the data variables.
void accept(String n)	: to assign n to text where the value of the parameter should be in lowercase.
void checkandfreq()	: to count the frequency of and.
void checkanfreq()	: to count the frequency of an.
void display()	: to display the frequency of “an” and “and” with suitable messages.

Specify class Frequency giving details of the constructor(), void accept(String), void checkand freq() and void display(). Also define the main function to create an object and call methods accordingly to enable the task. [ 10 ]

### 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:

<b>Class name</b>	:	DeciOct
<b>Data members</b>		
n	:	stores the decimal number
oct	:	stores the equivalent octal number
<b>Member functions:</b>		
DeciOct()	:	constructor to initialize the data members to 0
void getnum(int nn)	:	assigns 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 pseudo arithmetic sequences ( sequences that are in arithmetic progression).

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  $14 \times 3 = 42$

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

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

$14 \times 3 = 42$  [ middle element = 7 ]

A class pseudoarithmetic determines whether a given sequence is pseudo-arithmetic sequence. The details of the class are given below:-

<b>Class name</b>	:	Pseudoarithmetic
<b>Data members</b>		
n	:	to store the size of the sequence
a[ ]	:	integer array to store the sequence of numbers
ans, flag	:	to store the status
sum	:	to store the sum of sequence of numbers
r	:	to store the sum of 2 numbers
<b>Member functions</b>	:	
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 return false.

Specify the class Pseudoarithmetic, giving details of the constructor(), void accept(int) and Boolean check(). Also define the main function to create an object and call methods accordingly to enable the task. [ 10]

### Section C

**Answer any 2 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.

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:

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

void display() : displays the name and ranks along with the name having the topmost rank.

Specify the class Record giving details of the constructor(), void readvalues(), 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 class Stack are given below:

**Class name** : Stack  
st[] : the array to hold 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.

- Specify class Stack giving details of the constructors(), void pushname(String n), String popname() and void display(). The main function and algorithm need not be written.
- Under what principle does the above entity work?

### Question 13.

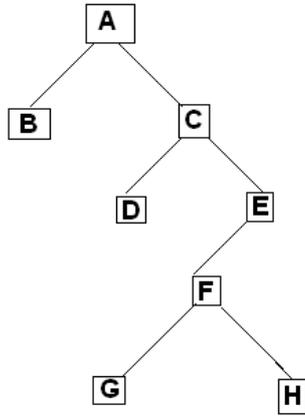
- A linked list is formed from the objects of the class,

```
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)
```

- Distinguish between worst-case and best-case complexity of an algorithm.
- Answer the following from the diagram of a Binary tree given below:



- i) Write the postorder tree traversal
- ii) Name the leaves of the tree
- iii) Height of the tree
- iv) Root of the tree