

PDF brought to you by ResPaper.com



## ISC Specimen 2011 Computer Science Paper 1 (Theory)

**Answer key / correct responses on:**

Click link: <http://www.respaper.com/isc/552/674.pdf>

Other papers by ISC : <http://www.respaper.com/isc/>

**Upload and share your papers and class notes on ResPaper.com. It is FREE!**

**ResPaper.com has a large collection of board papers, competitive exams  
and entrance tests.**

<http://www.respaper.com/>

# 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

- Draw the truth table to prove the propositional logic expression.  
 $a \Leftrightarrow b = (a \Rightarrow b) \cdot (b \Rightarrow a)$  [2]
- Determine if following wff is valid, satisfiable or unsatisfiable:  
 $(p \rightarrow q) \Rightarrow [\sim q \rightarrow (\sim p \wedge \sim q)]$  [2]
- Differentiate between a tautology and a contradiction. [2]
- Convert the following expression  $F(X,Y,Z) = XY + Y'Z$  into minterms. [2]
- Minimise  $F = AB + (AC)' + AB'C$  using Boolean laws. [2]

### Question 2

- What is dynamic binding? [2]
- Convert the following infix notation to postfix :  
 $A*(B/C)/((E*F)+G)$  [2]
- Define the terms Best-Case and Average-Case in complexities. [2]
- Each element of an array  $\text{arr}[15][20]$  requires 'W' bytes of storage. If the address of  $\text{arr}[6][8]$  is 4580 and the Base Address at  $\text{arr}[1][1]$  is 4000, find the width 'W' of each cell in the array  $\text{arr}[\ ][\ ]$  when the array is stored as **Column Major Wise**. [2]
- Define an abstract class. [2]

### Question 3

- (a) The following function **comb()** and **combi()** are a part of some class. Give the output when the function **combi()** is called. Show the dry run/working.

[5]

```

void combi ( )
{
    for (int i=0 ; i<5 ; i++)
    {
        for (int j=0 ; i<i+1 ; j++)
            System.out.print ( " " + comb ( i , j ) );
        System.out.println ( );
    }
}

long comb (int n , int k)
{
    long c = 1;
    for (int i = 1; i<=k ; i++ , n--)
        c = c*n/i;
    return c;
}

```

- (b) The following function **find()** and **perform()** are part of some class. Show the dry run / working:

```

int find ( int n, int p)
{
    if ( n == 0 )
        return p;
    else
        return find ( p % n , n );
}

void perform ( int m )
{
    int q = 14;
    int x = find ( q++, ++m);
    System.out.println(x);
}

```

- (i) What will the function **find (12, 8)** return? [2]  
 (ii) What will be the output of the function **perform()** when the value of **m** is 20. [2]  
 (iii) In one line state what the function **find()** is doing, apart from recursion. [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  $F(A,B,C,D) = \Sigma (0,2,6,8,10,11,14,15)$

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

(ii) Draw the Logic gate diagram of the reduced expression using NAND Gate only. [1]

(b) Given  $F(A,B,C,D) = \pi (5,7,8,10,12,14,15)$ .

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

(ii) Draw the Logic gate diagram of the reduced expression using NOR Gate only. [1]

#### Question 5

(a) Draw the logic diagram and Truth table to Encode the hexadecimal lines (A – F). Briefly explain the working of the logic diagram. [5]

(b) Simplify the equation and draw the gate for the reduced expression.

$$F = A'B + AB'C + A$$

[2]

(c) Define maxterms and minterms. Give one example of each. [3]

#### 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 S-O-P expression for the same. Reduce the S-O-P expression using K-map. [3]

(b) State the principal of duality. Give one example. [2]

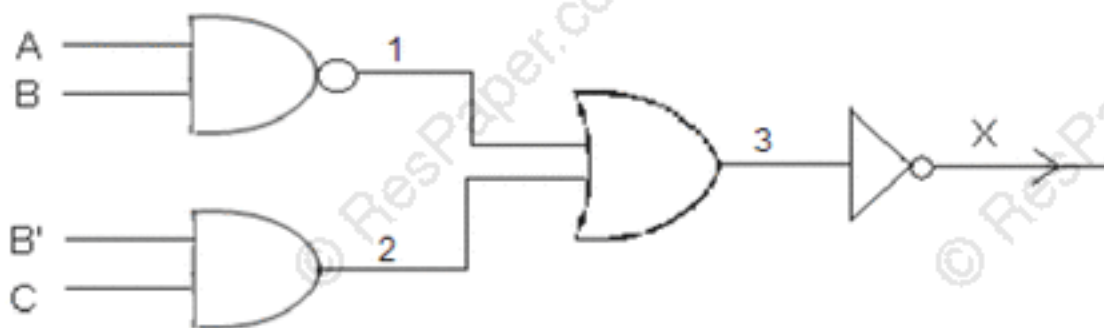
(c) How many select lines does an 8 : 1 multiplexer have? Briefly explain the working of a 4 : 1 multiplexer. Also draw the logic diagram of a 4 : 1 multiplexer. [5]



**Question 7**

- (a) How does Half adder differ from Full adders? Draw truth table for both the adders. Also derive expression for full adder and simplify the same. [5]

- (b) From the given logic diagram :



Derive Boolean expression and also draw the truth table. [3]

- (c) Verify the following :  
 $\Sigma (0,2,7,6) = \pi (1,3,4,5)$  [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 8**

[10]

Design a class **Prime** to fill an array of order [m x n] where the maximum value of both m and n is 20, with the first [m x n] prime numbers Row wise .The details of the members of the class are given below :

**Class name** : **Prime**

**Data members /instance variables** :

arr[ ][ ]	: Two dimensional integer array.
r	: integer to store the number of rows.
c	: integer to store the number of columns.

**Member functions :**

Prime( )	:	to accept the size of the array .
int isprime( int p )	:	return 1 if number is prime and 0 if not prime .
void fill ( )	:	to fill the elements of the array with the first (m x n) prime numbers.
void display( )	:	displays the array in a matrix form.

Specify the class **Prime** giving details of the **constructor** and member functions **int isprime (int)**, **void fill( )** and **void display( )** with **main( )** function to create an object and call the function accordingly.

**Question 9**

[10]

Design a class **Alpha** which enables a word to be arranged in ascending order according to its alphabets.

The details of the members of the class are given below :

**Class name** : **Alpha**

**Data members /instance variables:**

Str : to store a word

**Member functions :**

Alpha( )	:	default constructor
void readword( )	:	to accept the inputted word
void arrange ( )	:	to arrange the word in alphabetical order using any standard sorting technique.
void disp( )	:	displays the word .

Specify the class **Alpha** giving details of the **constructor** and the member functions **void readword( )**, **void arrange( )**, **void disp ( )** and defining the **main ( )** function to create an object and call the function in order to execute the class by displaying the original word and the changed word with proper message.

**Question 10**

[10]

The Combination function  $C(n, k)$  gives the number of different (unordered)  $K$  – elements Subsets that can be found in a given set of  $n$  elements. The function can be computed from the formula:

$$C(n, k) = \frac{n!}{k!(n-k)!}$$

Design a class **Combination** to implement this formula. Some of the data members and member functions are given below.

**Class name** : **Combination**

**Data members /instance variables :**

$n$  : integer number

$k$  : integer number

**Member functions :**

**Combination ( )** : to initialize the data members  $n = 0$  and  $k = 0$

**void read( )** : to accept the value of the data members

**int fact(int)** : return the factorial of a number using **recursion technique**.

**void compute( )** : calculate the combination value

**void display( )** : to show the result

Specify the class **Combination**, giving details of the **Constructor** and member functions **void read( )**, **int fact(int)**, **void compute( )** and **void display( )** with 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 / 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 )

**Programs to be written in Java.**

**The Algorithm must be written in general/standard form.**

### Question 11

[10]

A class Personal contains employee details and another class Retire calculates the employee's Provident Fund and Gratuity. The details of the two classes are given below:

**Class name** : **Personal**

#### **Data Members:**

Name	:	stores the employee name
Pan	:	stores the employee PAN number
basic_pay	:	stores the employee basic salary (in decimals)
acc_no	:	stores the employee bank account number

#### **Member functions:**

Personal( .... )	:	parameterized constructor to assign value to data members
void display( )	:	to display the employee details

**Class name** : **Retire**

#### **Data Members:**

Yrs	:	stores the employee years of service
Pf	:	stores the employee provident fund amount ( in decimals )
Grat	:	stores the employee gratuity amount ( in decimals )

#### **Member functions:**

Retire ( .... )	:	parameterized constructor to assign value to data members of both the classes.
void provident( )	:	calculates the PF as (2% of the basic pay) * years of service.
void gratuity( )	:	calculates the gratuity as 12 months salary, if the years of service is more than or equal to 10 years else the gratuity amount is nil.
void display1( )	:	Displays the employee details along with the PF (Provident Fund ) and gratuity amount.

Specify the class **Personal** giving details of the **constructor** and member functions **void display( )**. Using the **concept of inheritance**, specify the class **Retire** giving details of **constructor**, and the member functions **void provident( )**, **void gratuity( )** and the **void display1( )**. The main function need **not** be written.



**Question 12**

Chain is an entity which can hold at the most 50 integers. The chain enables the user to add and remove integers from both the ends i.e. front and rear. Define a class Chain with the following details:

**Class name** : **Chain**

**Data Members:**

ele[ ]	:	the 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:**

Chain(int max)	:	constructor to initialize the data cap = max, front = rear = 0 and to create the integer array.
void pushfront(int v)	:	to add integers from the front index if possible else display the message("full from front").
int popfront( )	:	to remove the return elements from front. If array is empty then return-999.
void pushrear(int v)	:	to add integers from the front index if possible else display the message("full from rear").
int poprear( )	:	to remove and return elements from rear. If the array is empty then return-999.

(a) Specify the class **Chain** giving details of the **constructor** and member function **void pushfront(int), int popfront( ), void pushrear(int) and int poprear ( )**. The main function need **not** be written.

[9]

(b) What is the common name of the entity described above?

[1]

**Question 13**

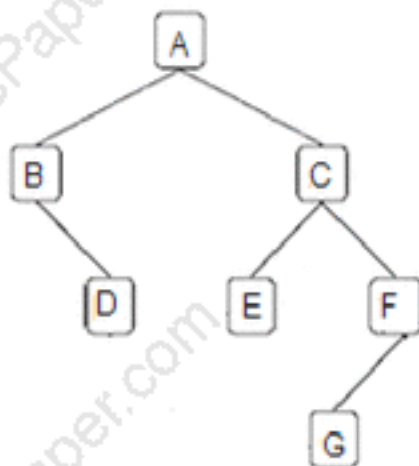
(a) Link Lists are linear data structure . Create Algorithms for the following operations :

(i) Insertion of a Node at the beginning in a Link List.

[4]

(ii) Deletion of the first Node in a Link list.

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



1. Name the leaves of the tree.

[1]

2. Write the path between A to G.

[1]

3. State the level of node E .

[1]

(c) For the given Binary Tree write the:

[3]

1. Postorder Sequence.

2. Inorder Sequence.

3. Preorder Sequence.

