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) State the two distributive laws of Boolean Algebra. Prove any one of them with the help of Truth Table. **[2]**

(b) Draw the truth table to verify the expression : **[2]**

$p \Rightarrow q$ is equivalent to $\sim q \Rightarrow \sim p$

($\sim q = \overline{q} = q'$)

(c) Find the complement of the following: **[2]**

$[(xy)' \cdot x]\ [(xy)' \cdot y]$

(d) Simplify the following Boolean Expression using laws of Boolean Algebra. At each step, clearly state the law used for simplification. **[2]**

$z \,.\, (z + x) \times (y + \overline{y})$

(e) Given **[2]**

$F(x, y, z) = xz + xy + yz$

Write the function in canonical sum of products form.

Question 2

(a) What do LIFO and FIFO stand for? **[2]**

(b) For an array of real numbers x [− 6… 8 , -12… 20] , find the address of x [5] [4], if x [1] [1] is stored in location 100 in the column major order. Assume that each element requires 4 bytes. **[2]**

(c) State the difference between an abstract class and an interface **[2]**

(d) Convert the following infix expression to its postfix form: **[2]**
b * [(a / d) - (c * (e - f))]

(e) Define a binary tree. **[2]**

Question 3

(a) The following function is a part of some class. It returns the value 1 when the number is an Armstrong number, otherwise it returns 0.

/* An Armstrong number is a number which is equal to the sum of the cube of its individual digits */

```
int arms ( int n )
{
   int digit = 0, sum = 0 ;
   int rem = n;
   while ( ? 1 ? )
    {
      digit = ? 2 ?;
      sum = sum + ? 3 ? ;
      rem = ? 4 ? ;
    }
   if (? 5 ? )
     return 1 ;
   else
     return 0 ;
}
```

(i) What is the expression/value at ? 1 ? **[1]**

(ii) What is the expression/value at ? 2 ? **[1]**

(iii) What is the expression/value at ? 3 ? **[1]**

(iv) What is the expression/value at ? 4 ? **[1]**

(v) What is the expression/value at ? 5 ? **[1]**

(b) Give output of the following function where x and y are arguments greater than 0. Show the dry run/working.

```
int strange (int x, int y)
{
//Assuming x>0 and y>0
  if(x>=y)
   {
      x = x-y;
      return strange(x , y);
   }
  else
      return x;
}
```

(i) What will the function strange(20,5) return ? **[2]**

(ii) What will the function strange(15,6) return ? **[2]**

(iii) In one line, state what the function is doing. **[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: **[5]**

$$F(P, Q, R, S) = \sum (0, 1, 3, 4, 5, 6, 7, 9, 10, 11, 13, 15)$$

Use Karnaugh's map to reduce the function F, using the SOP form. Draw a logic gate diagram for the reduced SOP form. You may use gates with more than two inputs. Assume that the variable and their complements are available as inputs.

(b) Given the Boolean function : **[5]**

$$X(P, Q, R, S) = \pi(3, 8, 10, 12, 13, 14, 15)$$

Use Karnaugh's map to reduce this function X using the given POS form. Draw a logic gate diagram for the reduced POS form. You may use gates with more than two inputs. Assume that the variables and their complements are available as inputs.

Question 5

The main safe in the nationalized bank can be opened by means of a unique password consisting of three parts. Different parts of the password are held by the Chairman, Regional Manager, Bank Manager and Head cashier of the bank, respectively.

In order to open the safe, any one of the following conditions must be satisfied:

The password of the Chairman, together with passwords of any two other officials, must be entered.

OR

The password of all the three bank officials, excluding the chairman, must be entered.

The inputs are:

A : Denotes the Chairman's password.

B : Denotes the Regional Manager's password.

C : Denotes the Bank Manager's password.

D : Denotes the Head Casher's password.

Output:

X – Denotes that the safe can be opened.

[1 indicates Yes and 0 indicates No in all cases]

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

(b) Reduce X(A, B, C, D) using Karnaugh's map, if possible. **[5]**

Draw the logic gate diagram for the reduced SOP expression for X(A, B, C, D) using AND & OR gates. You may use gates with two or more inputs. Assume that the variables and their complements are available as inputs.

Question 6

(a) Draw the truth table and logic circuit diagram for a Decimal to Binary Encoder. **[5]**

(b) Given : $F(x, y, z) = \sum (1,3,7)$ **[2]**

Verify : $F(x, y, z) = \pi (0,2,4,5,6)$

(c) Simplify the following expression by using Boolean laws. Show the working and also mention the laws used : **[3]**

X'Y'Z' + XYZ' + XY'Z' + X'YZ'

Question 7

(a) Define Cardinal Form of an expression and Canonical Form of an expression. Give an example for each. **[3]**

(b) Which gate is equivalent to : **[3]**
(NOR) OR (XOR)

(c) Define a Half Adder. Draw the Truth Table and 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 perfect square is an integer which is the square of another integer. For example, 4, 9, 16 .. are perfect squares. Design a Class **Perfect** with the following description: **[10]**

Class name	:	**Perfect**
Data members/instance variables		
n	:	stores an integer number
Member functions:		
Perfect( )	:	default constructor
Perfect(int)	:	parameterized constructor to assign a value to 'n'
void perfect_sq()	:	to display the first 5 perfect squares larger than 'n' (if n = 15, the next 3 perfect squares are 16, 25, 36)
void sum_of()	:	to display all combinations of consecutive integers whose sum is equal to n. (the number n = 15 can be expressed as 1 2 3 4 5 4 5 6 7 8

Specify the class Perfect giving details of the **constructors, void perfect_sq( )** and **void sum_of().** Also define the main function to create an object and call methods accordingly to enable the task.

Question 9

A class RecFact defines a recursive function to find the factorial of a number. The details of the class are given below: **[10]**

Class name : **RecFact**

Data members/instance variables

n	:	stores the number whose factorial is required.
r	:	stores an integer

Member functions

RecFact( )	:	default constructor
void readnum( )	:	to enter values for 'n' and 'r'
int factorial(int)	:	returns the factorial of the number using the Recursive Technique.
Void factseries( )	:	to calculate and display the value of $\frac{n!}{r!*(n-r)!}$

Specify the class **RecFact** giving the details of the **constructor** and member functions **void readnum( ), int factorial(int)** and **void factseries( )**. Also define the main function to create an object and call methods accordingly to enable the task.

Question 10

In "Piglatin" a word such as KING is replaced by INGKAY , while TROUBLE becomes OUBLETRAY and so on . The first vowel of the original word becomes the start of the translation, any preceding letters being shifted towards the end and followed by AY. **[10]**

Words that begin with a vowel or which do not contain any vowel are left unchanged.

Design a class **Piglatin** using the description of the data members and member functions given below:

Class name : **Piglatin**

Data members /instance variables :

Txt	:	to store a word
len	:	to store the length

Member functions :

Piglatin( )	:	constructor to initialize the data mrmbers
void readstring( )	:	to accept the word input in UPPER CASE
void convert ( )	:	converts the word into its piglatin form and displays the word (changed or unchanged)
void consonant( )	:	counts and displays the number of consonants present in the given word.

Specify the class **Piglatin** giving the details of the **constructor, void readstring( ), void convert( )** and **void consonant( ).** Also define the main function to create an object and call methods 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 class Author contains details of the author and another class Book List contains details of the books written by him. The details of the two classes are given below: **[10]**

Class name	:	**Author**
Data members		
authorno	:	stores the author's number
name	:	stores the author's name
Member functions		
Author ( )	:	default constructor
Author (…)	:	parameterised constructor to assign values to author number and name
void show( )	:	to display the author's details

Class name	:	**Booklist**

Data members/instance variables

bookno	:	Long type variable to the store book number
bookname	:	stores the book name
price	:	float variable to store the price
edition	:	integer type variable to store the edition number

Member functions

Booklist (…)	:	parameterized constructor to assign values to data members of both the classes
void show( )	:	to display all the details

Specify the class **Author** giving details of the **constructors** and member function **void show( ).** Using the **concept of Inheritance**, specify the class **Booklist** giving details of the **constructor** and the member function **void show( )**. Also define the main function to create an object and call methods accordingly to enable the task.

Question 12

In a computer game, a vertical column and a pile of rings are displayed. The objective of the game is to pile up rings on the column till it is full. It can hold 10 rings at the most. Once the column is full, the rings have to be removed from the top till the column is empty and then the game is over. Define the class RingGame with the following details: **[10]**

Class name	:	**RingGame**

Data members/instance variables

ring [ ]	:	array to hold rings (integer)
max	:	integer to hold maximum capacity of ring array
upper	:	integer to point to the upper most element

Member functions

RingGame(int m)	:	constructor to initialize, max = m & upper to – 1.
void jump-in(int)	:	adds a ring to the top of the column, if possible. otherwise, displays a message "Column full. Start removing rings".

void jump-out( )	:	removes the ring from the top, if column is not empty. otherwise, outputs a message, "Congratulations. The game is Over".

Specify the class **RingGame** giving the details of the **constructor** and functions **void jump-in(int)** and **void jump-out( ).** Also define the main function to create an object and call methods accordingly to enable the task.

Question 13

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

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

Write the algorithm **OR** a method for inserting a node in the end of the list.

The method declaration is given below :

void insertnode(Node start)

(b) State the complexity for the following algorithms : **[3]**

(i) Linear Search

(ii) Binary Search

(iii) Selection Sort

(c) List the nodes in the tree given below using: **[3]**

(i) Preorder Traversal

(ii) Postorder Traversal

(iii) Inorder traversal