

COMPUTER SCIENCE

PAPER 1

(THEORY)

(Maximum Marks: 70)

(Time allowed: 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 **six** questions from Part-II, choosing **two** 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 (20 Marks)

Answer **all** questions.

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

Question 1

- (a) State Involution law and prove it with the help of a truth table. [1]
- (b) Show that $X \vee \sim(Y \wedge X)$ is a tautology. [1]
- (c) Find the dual of: [1]
- $$Y \cdot X + X' + 1 = 1$$
- (d) Write the maxterm and minterm, when the inputs are $A=0$, $B=1$, $C=1$ and $D=0$. [1]
- (e) Draw the logic circuit of a NAND gate using NOR gates only. [1]

Question 2

- (a) Define the term *fall through* condition with reference to switch() case. [2]
- (b) Convert the following infix expression to postfix form: [2]
- $$A + B / C * (D / E * F)$$
- (c) A matrix $A[m][n]$ is stored with each element requiring 4 bytes of storage. If the base address at $A[1][1]$ is 1500 and the address at $A[4][5]$ is 1608, determine the number of rows of the matrix when the matrix is stored in **Column Major Wise**. [2]

This Paper consists of 9 printed pages and 1 blank page.

1216-868 A

© Copyright reserved.

Turn over

- (d) From the class declaration given below, state the nature of the identifiers A, B, C and D: [2]

class A extends B implements C, D

- (e) State *one* advantage and *one* disadvantage of using *recursion* over *iteration*. [2]

Question 3

The following function **Check()** is a part of some class. What will the function **Check()** return when the values of both 'm' and 'n' are equal to 5? Show the dry run / working. [5]

```
int Check (int m, int n)
{
    if (n == 1)
        return -- m --;
    else
        return ++ m + Check (m, --n);
}
```

PART – II (50 Marks)

Answer six questions in this part, choosing two questions from Section A, two from Section B and two from Section C.

SECTION - A

Answer any two questions.

Question 4

- (a) Given the Boolean function $F(A, B, C, D) = \Sigma (1,3,5,7,8,9,10,11,14,15)$.

- (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:

$$F(A, B, C, D) = \pi(4,6,7,10,11,12,14,15).$$

- (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) What is a *decoder*? Draw the logic diagram for a binary to octal (3 to 8) decoder. [3]
- (b) How is a *half adder* different from a *full adder*? Draw the truth table and derive the SUM and CARRY expression for a full adder. Also, draw the logic diagram for a full adder. [4]
- (c) State whether the following expression is a Tautology, Contradiction or a Contingency, with the help of a truth table: [3]

$$(X \Rightarrow Z) \vee \sim [(X \Rightarrow Y) \wedge (Y \Rightarrow Z)]$$

Question 6

- (a) A passenger is allotted a window seat in an aircraft, if he/she satisfies the criteria given below: [5]
- The passenger is below 15 years and is accompanied by an adult.
- OR**
- The passenger is a lady and is not accompanied by an adult.
- OR**
- The passenger is not below 15 years, but is travelling for the first time.

The inputs are:

INPUTS	
A	The passenger is below 15 years age.
C	The passenger is accompanied by an adult.
L	The passenger is a lady.
F	The passenger is travelling for the first time.

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

Output : **W** – Denotes the passenger is allotted a window seat (1 indicates yes and 0 indicates no)

Draw the truth table for the inputs and outputs given above and write the SOP expression for **W(A,C,L,F)**.

- (b) State the complement properties. Find the complement of the following Boolean expression using De Morgan's law: [3]
- $$AB' + A' + BC$$
- (c) Differentiate between *Canonical form* and *Cardinal form* of expression. [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 7

A disarium number is a number in which the sum of the digits to the power of their respective position is equal to the number itself.

[10]

Example: $135 = 1^1 + 3^2 + 5^3$

Hence, 135 is a disarium number.

Design a class **Disarium** to check if a given number is a disarium number or not. Some of the members of the class are given below:

Class name : **Disarium**

Data members/instance variables:

int num : stores the number
int size : stores the size of the number

Methods/Member functions:

Disarium(int nn) : parameterized constructor to initialize the data members $n = nn$ and $size = 0$
void countDigit() : counts the total number of digits and assigns it to size
int sumofDigits(int n, int p) : returns the sum of the digits of the number(n) to the power of their respective positions(p) using **recursive technique**
void check() : checks whether the number is a disarium number and displays the result with an appropriate message

Specify the class **Disarium** giving the details of the **constructor()**, **void countDigit()**, **int sumofDigits(int, int)** and **void check()**. Define the **main()** function to create an object and call the functions accordingly to enable the task.

Question 8

A class **Shift** contains a two dimensional integer array of order ($m \times n$) where the maximum values of both m and n is 5. Design the class **Shift** to shuffle the matrix (i.e. the first row becomes the last, the second row becomes the first and so on). The details of the members of the class are given below: [10]

Class name : **Shift**

Data member/instance variable:

mat[][] : stores the array element
m : integer to store the number of rows
n : integer to store the number of columns

Member functions/methods:

Shift(int mm, int nn) : parameterized constructor to initialize the data members $m = mm$ and $n = nn$
void input() : enters the elements of the array
void cyclic(Shift P) : enables the matrix of the object(P) to shift each row upwards in a cyclic manner and store the resultant matrix in the current object
void display() : displays the matrix elements

Specify the class **Shift** giving details of the **constructor()**, **void input()**, **void cyclic(Shift)** and **void display()**. Define the **main()** function to create an object and call the methods accordingly to enable the task of shifting the array elements.

Question 9

A class **ConsChange** has been defined with the following details:

[10]

Class name	:	ConsChange
Data members/instance variables:		
word	:	stores the word
len	:	stores the length of the word
Member functions/methods:		
ConsChange()	:	default constructor
void readword()	:	accepts the word in lowercase
void shiftcons()	:	shifts all the consonants of the word at the beginning followed by the vowels (e.g. spoon becomes spnoo)
void changeword()	:	changes the case of all occurring consonants of the shifted word to uppercase, for e.g. (spnoo becomes SPNoo)
void show()	:	displays the original word, shifted word and the changed word

Specify the class **ConsChange** giving the details of the **constructor()**, **void readword()**, **void shiftcons()**, **void changeword()** and **void show()**. 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 10

A super class **Bank** has been defined to store the details of a customer. Define a sub-class **Account** that enables transactions for the customer with the bank. The details of both the classes are given below: [5]

Class name : **Bank**

Data member/instance variable:

name : stores the name of the customer
accno : stores the account number
p : stores the principal amount in decimals

Member functions/methods:

Bank(...) : parameterized constructor to assign values to the instance variables
void display() : displays the details of the customer

Class name: **Account**

Data member/instance variable:

amt : stores the transaction amount in decimals

Member functions/methods:

Account(...) : parameterized constructor to assign values to the instance variables of both the classes
void deposit() : accepts the amount and updates the principal as $p=p + \text{amt}$
void withdraw() : accepts the amount and updates the principal as $p=p-\text{amt}$
If the withdrawal amount is more than the principal amount, then display the message "INSUFFICIENT BALANCE". If the principal amount after withdrawal is less than 500, then a penalty is imposed by using the formula $p=p-(500-p)/10$
void display() : displays the details of the customer

Assume that the super class **Bank** has been defined. Using the **concept of Inheritance**, specify the class **Account** giving details of the **constructor(...)**, **void deposit()**, **void withdraw()** and **void display()**.

The super class and the main function need not be written.

Question 11

A bookshelf is designed to store the books in a stack with LIFO (Last In First Out) operation. Define a class **Book** with the following specifications: [5]

Class name : **Book**

Data members/instance variables:

name[] : stores the names of the books
point : stores the index of the topmost book
max : stores the maximum capacity of the bookshelf

Methods/Member functions:

Book(int cap) : constructor to initialise the data members
max = cap and point = -1
void tell() : displays the name of the book which was last entered in the shelf. If there is no book left in the shelf, displays the message "SHELF EMPTY"
void add(String v) : adds the name of the book to the shelf if possible, otherwise displays the message "SHELF FULL"
void display() : displays all the names of the books available in the shelf

Specify the class **Book** giving the details of **ONLY** the functions **void tell()** and **void add(String)**. Assume that the other functions have been defined.

The main function need not be written.

Question 12

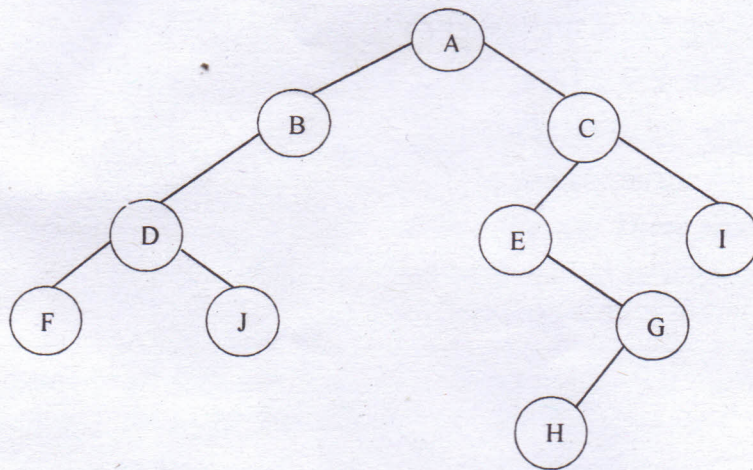
- (a) A linked list is formed from the objects of the class Node. The class structure of the Node is given below: [2]

```
class Node
{
    String name;
    Node next;
}
```

Write an *Algorithm* **OR** a *Method* to search for a given name in the linked list. The method of declaration is given below:

boolean searchName(Node start, String v)

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



- (i) Write the inorder traversal of the above tree structure. [1]
- (ii) Name the parent of the nodes B and G. [1]
- (iii) Name the leaves of the right sub-tree. [1]