COMPUTER SCIENCE
PAPER 1
(THEORY)

G2000370

Maximum Marks: 70
Time Allotted: Three Hours
Reading Time: Additional Fifteen minutes

Instructions to Candidates

1. You are allowed an **additional fifteen minutes** for **only** reading the question paper.
2. You must **NOT** start writing during the reading time.
3. This question paper has **15 printed pages and one blank page.**
4. It is divided into **two parts: Part I** and **Part II.**
5. It has **11 questions** in all.
6. **Part I** is compulsory and has **two** questions.
7. While attempting **Multiple Choice Questions** in **Part I**, you are required to **write only ONE option as the answer**.
8. **Part II** is divided into **three sections: A, B and C.**
9. **Each section in Part II** has **three questions. Any two** questions have to be attempted from each section.
10. The intended marks for questions are given in brackets [ ].

Instruction to Supervising Examiner

1. Kindly read **aloud** the Instructions given above to all the candidates present in the examination hall.

PART I – 20 MARKS

Answer all questions.

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

Question 1

(i) The complement of the Boolean expression $(P + Q') \cdot (R' + P)$ is: **[1]**

(a) $(P \cdot Q') + (R' \cdot P)$

(b) $(P' \cdot Q) + (R \cdot P')$

(c) $(P' \cdot Q) \cdot (R \cdot P')$

(d) $(P + Q') \cdot (R' + P)$

(ii) Given below are two statements marked Assertion and Reason. Read the two statements carefully and choose the correct option. **[1]**

Assertion: For overloaded method, signature matching happens at compile time.

Reason: The overloaded method must have a unique signature for its every prototype.

(a) Both Assertion and Reason are true and Reason is the correct explanation for Assertion.

(b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion.

(c) Assertion is true and Reason is false.

(d) Both Assertion and Reason are false.

(iii) According to the Principle of Duality, the Boolean equation $(X' + Y \cdot 0) \cdot X = 0$ will be equivalent to: **[1]**

(a) $(X + Y' \cdot 1) \cdot X' = 1$

(b) $(X \cdot Y' + 1) + X' = 1$

(c) $(X' \cdot Y + 1) + X = 1$

(d) $(X' \cdot Y + 0) + X = 0$

(iv) Commutative law states that: [1]

(a) $A \cdot (A + B) = A$

(b) $(A \cdot B) \cdot C = A \cdot (B \cdot C)$

(c) $A + (B + C) = (A + B) + C$

(d) $(A + B) = (B + A)$

(v) The canonical expression for $F(x, y, z) = \sum(1, 3, 6)$ is: [1]

(a) $(x \cdot y \cdot z') + (x \cdot y' \cdot z') + (x' \cdot y' \cdot z)$

(b) $(x + y + z') \cdot (x + y' + z') \cdot (x' + y' + z)$

(c) $(x' + y' + z) \cdot (x' + y + z) \cdot (x + y + z')$

(d) $(x' \cdot y' \cdot z) + (x' \cdot y \cdot z) + (x \cdot y \cdot z')$

(vi) Given below are two statements marked Assertion and Reason. Read the two statements carefully and choose the correct option. [1]

Assertion: An abstract class can contain abstract as well as non-abstract methods.

Reason: Abstract classes are meant to provide complete implementation of all types of methods in its sub class.

(a) Both Assertion and Reason are true and Reason is the correct explanation for Assertion.

(b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion.

(c) Assertion is true and Reason is false.

(d) Both Assertion and Reason are false.

(vii) Study the given propositions and the statements marked Assertion and Reason that follow it. Choose the correct option on the basis of your analysis. [1]

X – Sujata is a topper

Y – Sujata is in the merit list

Assertion: If Sujata is in the merit list, then she is a topper (Y => X).

Reason: Inverse is formed when both antecedent and consequent are negated.

(a) Both Assertion and Reason are true and Reason is the correct explanation for Assertion.

(b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion.

(c) Assertion is true and Reason is false.

(d) Both Assertion and Reason are false.

(viii) Consider the following code segment. [1]

class Car extends Vehicle implements Fourwheeler

{.....}

Which of the following statements are valid for the given code?

I *Vehicle* is a super class and *Fourwheeler* is an interface.

II Interface cannot have abstract methods.

III Keyword *extends* applies to class and keyword *implements* applies to interface.

(a) Only I and II

(b) Only II and III

(c) Only I and III

(d) Only II

(ix) Differentiate between Tautology and Contradiction. [1]

(x) For the given code segment, write Big O notation for worst case complexity. **[1]**

```
int i=1, j=1, k=1, p;
while (i <= p)
{
    statements
}
while (j<= p)
{
    while (k<= p)
    {
    statements
    }
}
```

Question 2

(i) Convert the following *infix* notation to *postfix* form. **[2]**

(A*B^C) + (D*E) where B^C = B^C

(ii) A matrix P[−10....10, 1...10] is stored in the memory with each element requiring 2 bytes of storage. If the base address is 2000, find the address of P[5][6] when the matrix is stored **Row Major wise.** **[2]**

(iii) The following function *think( )* is part of some class. Assume *f* is the lower bound and *l* is the upper bound of *arr[ ]*.

Answer the questions given below along with the dry run / working.

```
int think(int arr[ ], int f, int l)
{
  if (f > l)
    return 0;
  else
  {
  int current = (arr[f] % 2 == 0) ? arr[f] : 0;
  return current + think (arr, f + 1, l);
  }
}
```

(a) What will the function **think( )** return, if **arr[ ] = {9,7,12,16,19,25}, f = 0** and **l = 5** ? [2]

(b) What is the function **think( )** performing apart from recursion? [1]

(iv) The following function **getPart( )** is a part of some class which is used to extract **h** number of characters starting from index **b** of a string **str**. There are some places in the code marked by **?1?,?2?,?3?** which must be replaced by an expression / statement so that the function works correctly.

```
String getPart(String str, int b, int h)
{
  int beg= ?1?;
  int end =b+h-1;
  String g="";
  while(beg<=?2? && beg>=0 && end<str.length() )
  {
    g+=str.charAt(beg);
    ?3?;
  }
  return g;
}
```

(a) What is the expression or statement at ?1? [1]

(b) What is the expression or statement at ?2? [1]

(c) What is the expression or statement at ?3? [1]

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 3

(i) A food delivery app offers free home delivery to its customers who meet any of the following criteria. [5]

- The order is above ₹ 1000 and payment is made through UPI

OR

- Food is ordered from a partner restaurant and payment is made through UPI

OR

- The customer uses the app for the first time and places order above ₹ 1000

The inputs are:

INPUTS	
A	Order is above ₹ 1000
U	Payment is done through UPI
P	Food is ordered from a partner restaurant
F	Customer uses the app for the first time

(In all the above cases, 1 indicates YES, 0 indicates NO)

Output: **D** - Denotes free home delivery [1 indicates YES and 0 indicates NO in all cases]

Draw a truth table for the inputs and the outputs given above. Write the **SOP** expression for **D(A, U, P, F)**.

(ii) Reduce the above expression **D(A, U, P, F)** by using 4-variable Karnaugh map, showing the various groups (i.e., octal, quads and pairs). [5]

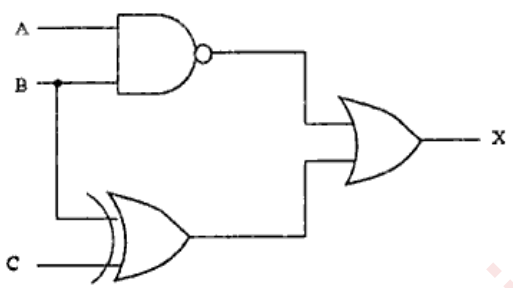
Draw the logic gate diagram for the reduced expression using NAND gates only. Assume that the variables and their complements are available as inputs.

Question 4

(i) (a) Reduce the Boolean function **F(X, Y, Z, W)** = $\pi(0,1,2,3,4,6,8,9,11,12)$ by using 4-variable Karnaugh map, showing the various groups (i.e., octal, quads and pairs). **[4]**

(b) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. **[1]**

(ii) From the logic gate diagram given below:



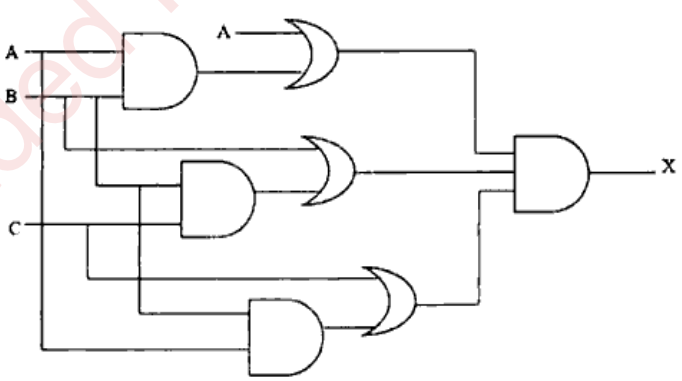
(a) Derive Boolean expression for X(A, B, C) and draw the truth table. **[4]**

(b) Write the canonical expression for SUM and CARRY of a half adder. **[1]**

Question 5

(i) Draw the logic gate diagram for an octal to binary encoder. State how multiplexer can be derived from a decoder. Mention *one* application of multiplexer. **[5]**

(ii) From the logic gate diagram given below, derive the Boolean expression for X(A, B, C) and reduce it using Boolean laws. **[3]**



(iii) Verify if the following compound proposition is **valid or not** using the **truth table.** **[2]**

$(A \wedge B) \wedge C \Rightarrow (A \wedge B \wedge C) = (A \wedge B) \Leftrightarrow (A \wedge B \wedge C)$

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 6 **[10]**

A class **Mighty** has been defined to create m1 and m2 as mighty numbers from integers n1 and n2. m1 will become mighty by placing the eventual sum of digits of n2 at the end of n1. m2 will become mighty by placing the eventual sum of digits of n1 at the end of n2.

Example: If

n1 = 235, the eventual sum of digits of n1 = 1

n2 =106, the eventual sum of digits of n2 = 7

then, m1= 2357

m2 = 1061

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

Class name	:	**Mighty**
Data members/instance variables:		
n1	:	to store first integer
n2	:	to store second integer
m1	:	to store the first mighty number
m2	:	to store the second mighty number
Methods/Member functions:		
Mighty( )	:	constructor to initialise data members with legal initial values
void accept( )	:	to accept values for n1 and n2
int sumofdigits(int x)	:	to return the eventual sum of digits of x using **recursive technique**
void calMighty( )	:	to calculate mighty numbers m1 and m2 by invoking **sumofdigits()**
void display( )	:	to display mighty numbers m1 and m2

Specify the class **Mighty** giving details of the **constructor( ),void accept( ), int sumofdigits(int), void calMighty( )** and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

Question 7 **[10]**

A class **JoinArray** contains an array of integer elements. Form a new array which will contain elements of the second array from last index to first index followed by elements of the first array from last index to first index.

Example:

Array1

9	7	6	4

Array2

8	3	2	1

Resultant Array

1	2	3	8	4	6	7	9

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

Class name	:	**JoinArray**
Data members/instance variables:		
arr[ ]	:	integer array
size	:	integer to store the size of the array
Member functions/methods:		
JoinArray(int s)	:	parameterised constructor to initialise size = s
void accept( )	:	to input the elements of the array
JoinArray attach(JoinArray P, JoinArray Q)	:	to return the resultant array after combining arrays of objects P and Q as per the description given above
void display( )	:	to display the array elements

Specify the class **JoinArray** giving details of the **constructor(int)**, **void accept( )**, **JoinArray attach(JoinArray, JoinArray)** and **void display( )**. Define the **main( )** function to create objects and call the functions accordingly to enable the task.

Question 8 [10]

A class **Palprime** has been defined to display **Palprime** words from a sentence. Palprime is a palindrome word and its length is a prime number.
Example:

Input: MY MOM POSSESSES A GOOD LEVEL OF KNOWLEDGE IN MALAYALAM

Output: MOM, LEVEL

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

Class name	:	**Palprime**
Data member/instance variable:		
line	:	to store a sentence
Methods/Member functions:		
Palprime( )	:	constructor to initialise data member
void input( )	:	to accept the sentence in UPPER CASE
boolean isPalin(String n)	:	to check whether the string n is a palindrome word and return true, otherwise return false
boolean primelen(String n)	:	to find the length of the string and to check for prime and return true if the length is a prime number otherwise return false
void display( )	:	to display the palprime word(s) from the sentence by invoking isPalin() and primelen()

Specify the class **Palprime** giving details of the **constructor( ), void input( ), boolean isPalin(String), boolean primelen(String)** and **void display( )**. 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 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 9

A circular queue is a linear data structure that allows data insertion at the rear and removal from the front, with the rear end connected to the front end forming a circular arrangement.

Given below are the details of class **MerryGoRound**

Class name : **MerryGoRound**

Data members/instance variables:

q[ ]	:	an array to hold integers
cap	:	to store the maximum capacity of the array
front	:	to point the index of the front end
rear	:	to point the index of the rear end

Methods/Member functions:

MerryGoRound(int n)	:	parameterised constructor to initialise cap = n, front = rear = 0
void add(int val)	:	to add integers from the rear index if possible, else display the message "QUEUE IS FULL"
int remove( )	:	to remove and return the integer from front if any, else return –999
void display()	:	to display the elements of the circular queue in the order of front to rear

(i) Specify the class **MerryGoRound** giving details of the functions **void add(int)** and **int remove( )**. Assume that the other functions have been defined. **[4]**

The main() function and algorithm need NOT be written.

(ii) State the principle on which the entity works. **[1]**

Question 10 [5]

A superclass **Train** has been defined to store the details of a train for booking tickets. Define a subclass **Booking** to compute the total cost of the tickets based on the number of passengers and any discount for group bookings.
The details of the members of both the classes are given below:

Class name	:	**Train**
Data members/instance variables:		
trainName	:	to store the name of the train
trainNumber	:	to store the train number
ticketPrice	:	to store the price of a single ticket in decimal
Methods/Member functions:		
Train(…)	:	parameterised constructor to assign values to data members
void displayDetails( )	:	to display the train details
Class name	:	**Booking**
Data members/instance variables:		
numPassengers	:	to store the number of passengers for booking
groupDiscount	:	to store the group discount percentage
totalCost	:	to store the total cost after applying the group discount in decimal
Methods/Member functions:		
Booking(...)	:	parameterised constructor to assign values to data members of both the classes
void calCost( )	:	to calculate the cost as per the formula (ticket price * numPassengers), apply a group discount of 10% if numPassengers is greater than 5 and calculate the total cost with discount, if any
void displayDetails( )	:	to display the train details, number of passengers, group discount and the total cost with discount, if any

Assume that the superclass Train has been defined. Using the **concept of Inheritance**, specify the class **Booking** giving the details of the **constructor(…), void calCost( )** and **void displayDetails( )**.
The superclass, main function and algorithm need NOT be written.

Question 11

(i) A linked list is formed from the objects of class **VotersList**. The class structure is given below. **[2]**

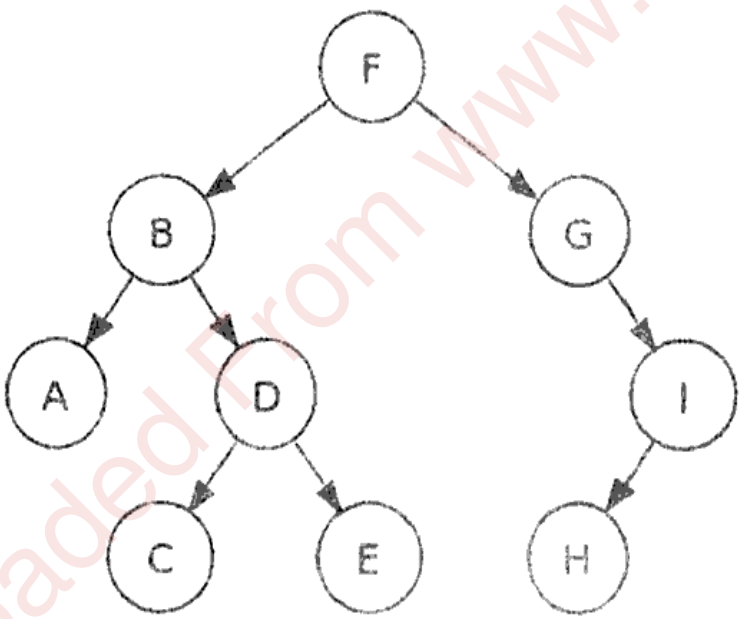
```
class VotersList
{
        int age;
        VotersList link;
}
```

Write an *Algorithm* **OR** a *Method* to count and return total number of nodes whose age > = 60.

The method prototype is as follows:

int countNodes(VotersList start)

(ii) Refer to the Binary Tree given below and answer the questions that follow:



(a) Write the post-order traversal of the above tree structure. [1]

(b) Name the internal nodes of the right subtree and the external nodes of the left subtree [1]

(c) Write the size of the tree and the degree of node D. [1]