

COMPUTER SCIENCE PAPER 1 (THEORY)

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 **12 printed pages**.
4. It is divided into **two parts: Part I and Part II**.
5. It has **eleven 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, briefly indicate your working and reasoning, wherever required.

Question 1

[1]

(i) According to the Principle of Duality, the Boolean equation $Q' \cdot 0 + P' \cdot Q' + P' \cdot Q = P'$ will be equivalent to:

- (a) $Q \cdot 0 + P \cdot Q + P \cdot Q' = P$
- (b) $Q' \cdot 1 + P' \cdot Q' + P \cdot Q' = P'$
- (c) $(Q' + 1) \cdot (P' + Q') \cdot (P' + Q) = P'$
- (d) $(Q' + 0) \cdot (P' + Q') \cdot (P' + Q) = P'$

(ii) Consider the following statement written in class Student where school_Name is its data member. [1]

```
static final String school_Name = "Co-Ed School";
```

Which of the following statements are valid for school_Name?

- I. All objects of class Student share the same value of school_Name.
- II. The value of school_Name cannot be changed during program execution.
- III. The keywords static and final cannot be used together for a variable.

- (a) Only I and II
- (b) Only II and III
- (c) Only I and III
- (d) Only III

(iii) Study the given propositions and the statements marked Assertion and Reason that follow. Choose the correct option based on your analysis. [1]

P = You practise regularly

Q = You become skilled

S1 = $P \Rightarrow Q$

S2 = $\sim P \vee Q$

Assertion: S1 and S2 are logically equivalent.

Reason: A conditional statement $P \Rightarrow Q$ can be expressed as $\sim P \vee Q$

- (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.

- (iv) The Boolean equations $a + 1 = a$ and $a \cdot 0 = 0$ correspond to: [1]
- (a) Involution Law
 - (b) Law of Identity
 - (c) Distributive Law
 - (d) Law of Complements

- (v) The *worst case* complexity for following code segment is: [1]

```
for(int i=1; i<= n; i++)
{
    for(int j=1; j<= i; j++)
    {
        statement;
    }
}
```

- (a) $O(n+i)$
 - (b) $O(n \times i)$
 - (c) $O(n)$
 - (d) $O(n^2)$
- (vi) Given below are two statements marked, Assertion and Reason. Read the two statements carefully and choose the correct option. [1]

Assertion: An interface in Java contains abstract and non-abstract methods.

Reason: All methods in an interface must be implemented by any class that extends this interface.

- (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) The complement of the Boolean expression $a \cdot b' + a' + a' \cdot b$ is: [1]
- (a) $a + b' \cdot a' \cdot a' + b$
 - (b) $(a' + b) \cdot a \cdot (a + b')$
 - (c) $(a + b') \cdot a' \cdot (a + b')$
 - (d) $a' \cdot b + a + a \cdot b'$

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

Assertion: The return statement enables the exit of the program; control from the current method.

Reason: If a method's return type is void, it can still contain return 0 statement to return nothing.

- (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.
- (ix) Consider the two propositions given below: [1]
A = You use ecofriendly methods
B = Pollution is reduced
If A implies to B, then write its *Contrapositive* statement.
- (x) What is *Gray code* in Karnaugh map? [1]

Question 2

- (i) Convert the following *infix* notation to *prefix* form. [2]
 $P / Q + (S * F + X / R)$
- (ii) A matrix $G[3...7, 2...5]$ is stored in the memory, with each element requiring 4 bytes of storage. If the address of $G[5][4]$ is 6000, find the **base address** when the matrix is stored **Column Major Wise**. [2]
- (iii) The following function `workOut()` is a part of some class. Assume 'n' is a positive integer.

```
String workOut (int n)
{
    if (n == 0)
        return "";
    int rem = n % 16;
    char cr = (rem < 10) ? (char)(rem + '0') : (char)(rem - 10 + 'A');
    return workOut (n / 16) + cr;
}
```

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

- (a) What will the function `workOut(220)` return? [2]
- (b) What is the function `workOut()` performing apart from recursion? [1]

- (iv) The following function `isTech()` is a part of some class which is used to check if a given number is a Tech number or not. There are some places in the code marked by ?1?, ?2?, ?3? which may be replaced by a statement / expression so that the function works properly.

A number is **Tech** number if the count of digits is even and the square of the sum of its two equal halves is equal to the number itself.

Example: $2025 = 20 + 25 = (45)^2 = 2025$

```
boolean isTech(int n)
{
    String s = String.valueOf(n);
    int len = ?1?;
    if (len % 2 != 0)
        return ?2?;
    int first = Integer.parseInt(s.substring(0, len / 2));
    int second = Integer.parseInt(s.substring(len / 2));
    int sum = first + second;
    return ?3? == n;
}
```

What are the expressions or statements at ?1?, ?2? and ?3?

[3]

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) What is a *decoder*? [1]
- (b) Draw the logic gate diagram for decoding the binary numbers {0011, 0100, 0101, 0111, 1011, 1110} to hexa-decimal numbers. [3]
- (c) State *any one* application of a *multiplexer*. [1]
- (ii) The Chain rule states that $[(a \Rightarrow b) \cdot (b \Rightarrow c)] \Rightarrow (a \Rightarrow c)$. Prove this rule using Boolean laws. [3]
- (iii) Given that $P = 0, Q = 1, R = 0, S = 0$, write its:
- (a) Maxterm [1]
- (b) Minterm [1]

Question 4

- (i) According to the ancient laws of the Valley of Peace, a candidate can become the Dragon Warrior only if they satisfy any one of the following conditions: [5]

- The candidate belongs to the Panda Clan and has been trained for more than 5 years under the Grand Master

OR

- The candidate possesses the Secret Chi Power but does not belong to the Panda Clan

OR

- The candidate is recommended by the Grand Master, but neither belongs to the Panda Clan nor has been trained for more than 5 years

The inputs are:

INPUTS	
C	Belongs to the Panda Clan
T	Trained for more than 5 years
P	Possesses Secret Chi Power
R	Recommended by the Grand Master

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

Output: X - Denotes eligibility to become the Dragon Warrior

(1 = eligible and 0 = not eligible)

Draw the truth table for the inputs and outputs given above. Write the SOP expression for X(C, T, P, R).

- (ii) Construct the logic gate diagram for a Full Adder using two Half Adders. [3]

- (iii) Draw a truth table to verify if the following proposition is a Tautology, a Contradiction or a Contingency. [2]

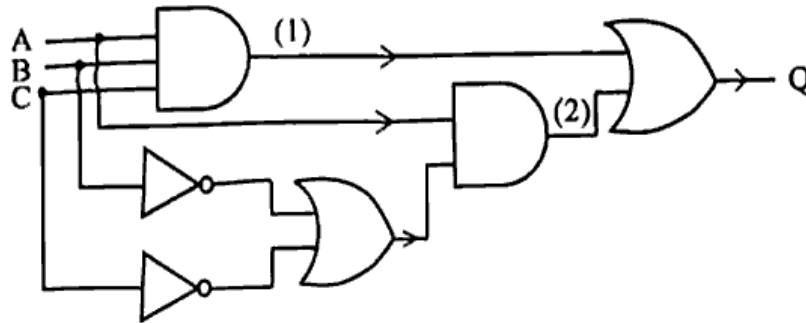
$$(A \wedge \sim B) \Rightarrow (\sim A \vee B)$$

Question 5

- (i) (a) Reduce the Boolean function $F(A,B,C,D) = \pi(0,1,2,3,7,8,9,10,11,12,13,15)$ 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, derive the Boolean expression for (1), (2) and Q. Reduce the derived expression. [3]



- (iii) Draw the logic gate diagram for 2-input AND gate using NOR gates only. [2]
Show the expression at each step.

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 **TimeOp** has been defined to add any two accepted time periods.

Example: Time A = 6 hours 35 minutes 40 seconds

Time B = 7 hours 45 minutes 30 seconds

Time A + Time B = 14 hours 21 minutes 10 seconds

(where 60 minutes = 1 hour and 60 seconds = 1 minute)

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

Class name : **TimeOp**

Data member/instance variable:

arr[] : integer array to hold three elements (hours, minutes and seconds)

Methods/Member functions:

TimeOp() : default constructor

void readTime() : to accept the elements of the array

TimeOp addTime(TimeOp tt) : to add the time of the parameterised object *tt* and the current object, to store it in a local object and return it

void dispTime() : to display the array elements in hours:minutes:seconds format

Specify the class **TimeOp** giving the details of the **constructor()**, **void readTime()**, **TimeOp addTime(TimeOp)** and **void dispTime()**. Define the **main()** function to create objects and call the functions accordingly to enable the task.

Question 7

- (i) A class **Trimorphic** has been defined to accept a positive integer from the user and display if it is a Trimorphic number or not.

[A number is said to be Trimorphic if the cube of the number ends with the number itself.]

Example 1: $24^3 = 13824$ ends with 24

Example 2: $5^3 = 125$ ends with 5

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

Class name : **Trimorphic**

Data members/instance variables:

n : to store the number
cube : to store the cube of the number

Methods/Member functions:

Trimorphic() : constructor to initialise the data members with legal initial values

void accept() : to accept a number

boolean check(int num, long c) : to compare *num* with the ending digits of *c* using recursive technique

void result() : to check whether the given number is a trimorphic number by invoking the function *check()* and to display an appropriate message

Specify the class **Trimorphic** giving the details of the **constructor()**, **void accept()**, **boolean check()** and **void result()**. Define the **main()** function to create an object and call the functions accordingly to enable the task. [8]

- (ii) State any two differences between *iteration* and *recursion*. [2]

Question 8

[10]

Design a class **PendulumS** to perform an operation on a word containing alphabets in upper case only. Rearrange the word by putting the lowest ASCII value character at the centre and the second lowest ASCII value character to its right and the third to its left and so on.

Example 1 : Input : COMPUTER

Output : TPMCEORU

Example 2 : Input : SCIENCE

Output : SIECCEN

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

Class name : **PendulumS**

Data members/instance variables:

wrd : to store the original word

newwrd : to store the rearranged word

Methods/Member functions:

PendulumS(String k) : parameterised constructor to initialise *wrd = k* and *newwrd = ""*

int minCharIndex(String str) : to find the index of the minimum ASCII value character in *str* and return it

void arrange() : to rearrange the characters of *wrd* as per the given instructions and store it in *newwrd* by invoking *minCharIndex()*

void display() : to display the original word and the rearranged word

Specify the class **PendulumS** giving the details of the **constructor()**, **int minCharIndex(String)**, **void arrange()** and **void display()**. 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 9

In any internet browser, a user can visit new webpages and go back to previously visited webpages. Each new webpage URL is stored in browser's memory such that when the user clicks 'Back' button, the previous webpage gets displayed.

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

Class name	: Browser
Data members/instance variables:	
pages[]	: an array to hold the URLs of visited webpages
max	: to store the maximum capacity of the array
top	: to point to the index of the last visited webpages
Methods/Member functions:	
Browser(int cap)	: constructor to assign $max = cap$ and $top = -1$
void visit(String url)	: to add URL of a new webpage if possible, else display the message "Browser history full"
String back()	: to remove and return the last visited webpage URL, if present, else to return the message "No previous browser history"

- (i) Specify the class **Browser** giving details of the functions **void visit(String)** and **String back()**. Assume that the other functions have been defined. [4]
- (ii) Name the entity described above and state its principle. [1]

Question 10**[5]**

A superclass **Hotel** has been defined to store the details of a hotel. Define a subclass **Customer** to calculate the total bill for a customer as per the following criteria:

Room type	Additional Amount
Executive	10% of room rent
Suite	20% of room rent

The details of the members of both the classes are given below:

Class name : **Hotel**

Data members/instance variables:

hname : to store hotel name

roomrent : to store the rent per day

Methods/Member functions:

Hotel(...) : parameterised constructor to assign values to its data members

void show() : to display hotel details

Class name : **Customer**

Data members/instance variables:

cname : to store customer name

days : to store the number of days of stay

type : to store the room type

surcharge : to store the additional amount

amt : to store the total amount

Methods/Member functions:

Customer(...) : parameterised constructor to assign values to data members of both the classes

void compute() : to calculate the surcharge based on the room type as given above. Also, to calculate the total amount as:

$-(\text{room rent} + \text{surcharge}) \times \text{days}$

void show() : to display hotel and customer details

Assume that the superclass **Hotel** has been defined. Using the concept of **Inheritance**, specify the class **Customer**, giving details of **constructor(...)**, **void compute()** and **void show()**

The super class, main function and algorithm need NOT be written.

Question 11

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

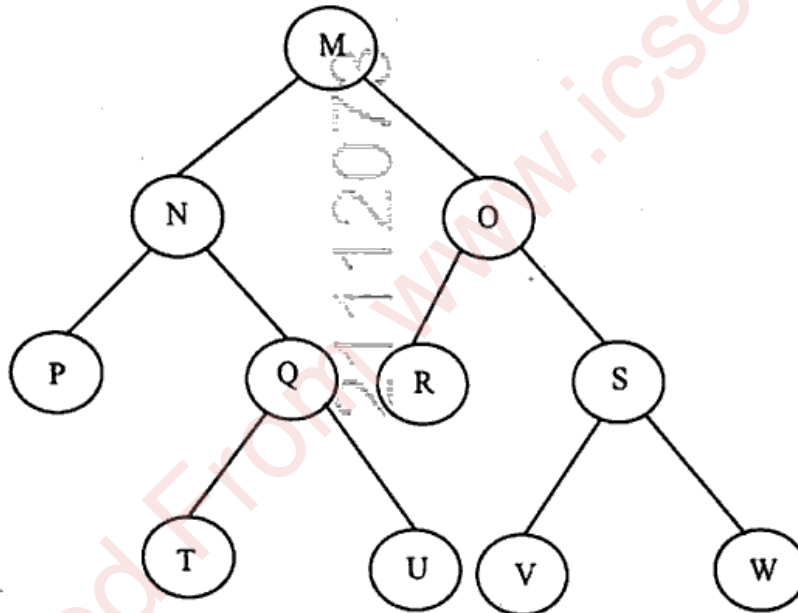
```
class Word
{
    String value;
    Word next;
}
```

Write an *Algorithm* OR a *Method* to count and display the number of nodes whose *value* starts with a consonant.

The method declaration is as follows:

```
void countConsonant(Word first)
```

- (ii) Answer the following questions based on the diagram of a Binary Tree given below:



- (a) Write the *pre-order* traversal of the above tree. [1]
- (b) State the level of Node Q, when the root is at level 0. [1]
- (c) State the size of the left subtree and the right subtree. [1]