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
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

(i) The law which states a + (b.c) = (a+b) . (a+c) is: **[1]**

(a) Associative Law

(b) Distributive Law

(c) Involution Law

(d) Commutative Law

(ii) The dual of (P + P') • (Q + 0) = Q is: **[1]**

(a) P.P' + Q.1 = Q

(b) P.P' + Q.0 = Q

(c) P.P + Q.1 = Q'

(d) P+P' + Q+1 = Q

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

(a) (P+Q).R

(b) PQR

(c) (P'+Q') .R'

(d) (P'+Q').R

(iv) If (x => ~y) then, its inverse will be: **[1]**

(a) x => y

(b) y => x

(c) ~y => x

(d) ~x => y

(v) Transitive nature of inheritance is implemented through **[1]**

(a) Single inheritance

(b) Multiple inheritance

(c) Hybrid inheritance

(d) Multilevel inheritance

(vi) Write the canonical sum of product form of the function y(A,B) = A + B. **[1]**

(vii) Name the basic gate that is equivalent to two NOR gates connected in series. **[1]**

(viii) State any one purpose of using the keyword *super* in Java programming. **[1]**

(ix) Define *Interface* with respect to data abstraction. **[1]**

(x) What is a *linked list*? **[1]**

Question 2

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

(P / Q – R) * (S + T)

(ii) A matrix N[11][8] is stored in the memory with each element requiring 2 bytes of storage. If the base address at N[2][3] is 2140, find the address of N[7][5] when the matrix is stored in Row Major Wise. **[2]**

(iii) With reference to the code given below answer the questions that follow.

```
void Solve(int n)
{  int a=1,b=1;
   for (int i=n;i>0;i=i/10)
  { int d=i%10;
    if (d%2==0)
       a=a*d;
    else
      b=b*d;
  }
  System.out.println(a+" "+b);
}
```

(a) What will the function **Solve( )** return when the value of **n=3269**? **[2]**

(b) What is the method **Solve( )** computing? **[1]**

(iv) The following function **quiz( )** is a part of some class. Assume 'n' is a positive integer, greater than 0. Answer the given questions along with dry run / working.

```
int quiz( int n)
 {
     if ( n <= 1 )
        return  n;
     else
         return  (--n % 2) + quiz(n/10);
 }
```

(a) What will the function **quiz( )** return when the value of **n=36922**? **[2]**

(b) State in one line what does the function **quiz( )** do, apart from recursion? **[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) Given the Boolean function **F(A,B,C,D) = ∑(0, 1, 2, 3, 4, 6, 9, 11, 13)**.

(a) Reduce the above expression 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) Given the Boolean function **F(A,B,C,D) = π(0, 1, 3, 5, 6, 7, 9, 11, 13, 14, 15)**.

(a) Reduce the above expression 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]**

Question 4

(i) A family intends to purchase a smart phone depending on the criteria given below: **[5]**

- Quad core processor with internal memory of 64 GB or more but not a resale phone

OR

- Resale phone with quad core processor but without warranty

OR

- Processor is not a quad core but with a warranty of 1 year and the internal memory is of 64 GB or more

The inputs are:

INPUTS	
P	Quad core processor
M	Internal memory of 64 GB or more
R	Resale phone
W	Warranty of 1 year

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

Output: X [1 indicates purchased, 0 indicates not purchased for all cases]

Draw the truth table for the inputs and outputs given above and write the **SOP** expression

(ii) What is a *half adder*? Draw the logic circuit for the SUM and CARRY expression of a half adder using only NAND gates. **[3]**

(iii) Simplify the following expression using Boolean laws: **[2]**

$F = PQ + (P + Q) \bullet (P + PR) + Q$

Question 5

(i) What is a *decoder*? How is it different from a multiplexer? Draw the logic circuit for a 2 to 4 decoder and explain its working. **[5]**

(ii) Verify if the following proposition is valid: **[3]**

$$(P \Rightarrow Q) \wedge (P \Rightarrow R) = P \Rightarrow (Q \wedge R)$$

(iii) Write the *maxterm* and *minterm* for the function F(A, B,C,D) when, A=1, B=1, C=0 and D=1. **[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 6

Design a class **Pronic** to check if a given number is a pronic number or not. [A number is said to be pronic if the product of two consecutive numbers is equal to the number] **[10]**

Example: $0 = 0 \times 1$

$2 = 1 \times 2$

$6 = 2 \times 3$

$12 = 3 \times 4$

thus, 0, 2, 6, 12... are pronic numbers.

Some of the members of the class are given below:

Class name	:	**Pronic**
Data members/instance variables:		
num	:	to store a positive integer number
Methods / Member functions:		
Pronic( )	:	default constructor to initialize the data member with legal initial value
void acceptnum( )	:	to accept a positive integer number
boolean ispronic(int v)	:	returns *true* if the number '*num*' is a pronic number, otherwise returns *false* using **recursive technique**
void check( )	:	checks whether the given number is a pronic number by invoking the function *ispronic()* and displays the result with an appropriate message

Specify the class **Pronic** giving details of the **constructor( )**, **void acceptnum( )**, **boolean ispronic(int)** and **void check( )**. Define a **main( )** function to create an object and call the functions accordingly to enable the task.

Question 7

Design a class **OddEven** to arrange two single dimensional arrays into one single dimensional array, such that the odd numbers from both the arrays are at the beginning followed by the even numbers. **[10]**

Example: Array 1: { 2, 13, 6, 19, 26, 11, 4 }
Array 2: { 7, 22, 4, 17, 12, 45 }

Arranged Array = { 13, 19 11, 7, 17, 45, 2, 6, 26, 4, 22, 4, 12 }

Some of the members of the class are given below:

Class name	:	**OddEven**
Data members/instance variables:		
a[ ]	:	to store integers in the array
m	:	integer to store the size of the array
Methods / Member functions:		
OddEven(int mm)	:	parameterised constructor to initialize the data member m=mm
void fillarray( )	:	to enter integer elements in the array
OddEven arrange(OddEven P, OddEven Q)	:	stores the odd numbers from both the parameterized object arrays followed by the even numbers from both the arrays and returns the object with the arranged array
void display( )	:	displays the elements of the arranged array

Specify the class **OddEven** giving details of the **constructor( )**, **void fillarray( )**, **OddEven arrange(OddEven, OddEven)** and **void display( )**. Define a **main( )** function to create objects and call the functions accordingly to enable the task.

Question 8 **[10]**

A class **Encrypt** has been defined to replace only the vowels in a word by the next corresponding vowel and forms a new word. i.e. A → E, E → I, I → O, O → U and U → A

Example: Input: COMPUTER

Output: CUMPATIR

Some of the members of the class are given below:

Class name	:	**Encrypt**
Data members/instance variables:		
wrd	:	to store a word
len	:	integer to store the length of the word
newwrd	:	to store the encrypted word
Methods / Member functions:		
Encrypt( )	:	default constructor to initialize data members with legal initial values
void acceptword( )	:	to accept a word in UPPER CASE
void freqvowcon( )	:	finds the frequency of the vowels and consonants in the word stored in '*wrd*' and displays them with an appropriate message
void nextVowel( )	:	replaces only the vowels from the word stored in '*wrd*' by the next corresponding vowel and assigns it to '*newwrd*', with the remaining alphabets unchanged
void disp( )	:	Displays the original word along with the encrypted word

Specify the class **Encrypt** giving details of the **constructor( )**, **void acceptword( )**, **void freqvowcon( ), void nextVowel( )** and **void disp( )**. Define a **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

Holder is a kind of data structure which can store elements with the restriction that an element can be added from the rear end and removed from the front end only.
The details of the class **Holder** is given below:

Class name	**:**	**Holder**

Data members/instance variables:

Q[ ]	:	array to hold integers
cap	:	maximum capacity of the holder
front	:	to point the index of the front end
rear	:	to point the index of the rear end

Methods / Member functions:

Holder(int n)	:	constructor to initialize cap=n, front= 0 and rear=0
void addint(int v)	:	to add integers in the holder at the rear end if possible, otherwise display the message " HOLDER IS FULL "
int removeint( )	:	removes and returns the integers from the front end of the holder if any, else returns –999
void show( )	:	displays the elements of the holder

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

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

(ii) Name the entity described above and state its principle. **[1]**

Question 10 **[5]**

A super class **Bank** has been defined to store the details of the customer in a bank. Define a subclass **Interest** to calculate the compound interest.

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

Class name	:	**Bank**
Data members/instance variables:		
name	:	to store the name of the customer
acc_no	:	integer to store the account number
principal	:	to store the principal amount in decimals
Methods / Member functions:		
Bank(...)	:	parameterized constructor to assign values to the data members
void display( )	:	to display the customer details
Class name		**Interest**
Data members/instance variables:		
rate	:	to store the interest rate in decimals
time	:	to store the time period in decimals
Methods / Member functions:		
Interest(...)	:	parameterized constructor to assign values to the data members of both the classes
double calculate( )	:	to calculate and return the compound interest using the formula [$CI = P (1 + R/100)^N - P$] where, P is the principal, R is the rate and N is the time
void display( )	:	to display the customer details along with the compound interest

*Assume that the super class **Bank** has been defined*. Using the **concept of inheritance**, specify the class **Interest** giving the details of the **constructor(...)**, **double calculate( )** and **void display( )**.

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: **[2]**

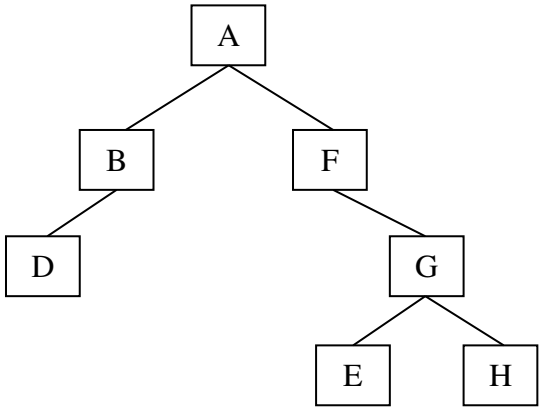
```
class Node
{
        int num;
        Node next;
}
```

Write an *Algorithm* **OR** a *Method* to insert a node at the beginning of an existing linked list.

The method declaration is as follows:

void InsertNode(Nodes starPtr, int n)

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



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

(b) Name the children of the nodes B and G. **[1]**

(c) State the root of the right sub tree. **[1]**