

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 the law represented by the following proposition and prove it with the help of a truth table: [1]

$$P \vee P = P$$

- (b) State the Principle of Duality. [1]

- (c) Find the complement of the following Boolean expression using De Morgan's law: [1]

$$F(a,b,c) = (b' + c) + a$$

- (d) Draw the logic diagram and truth table for a 2 input XNOR gate. [1]

- (e) If $(\sim P \Rightarrow Q)$ then write its: [1]

(i) Inverse

(ii) Converse

Question 2

- (a) What is an *interface*? How is it different from a *class*? [2]
- (b) Convert the following infix expression to postfix form: [2]
 $P * Q / R + (S + T)$
- (c) A matrix $P[15][10]$ is stored with each element requiring 8 bytes of storage. If the base address at $P[0][0]$ is 1400, determine the address at $P[10][7]$ when the matrix is stored in **Row Major Wise**. [2]
- (d) (i) What is the worst case complexity of the following code segment:
- ```
for (int x = 1; x <= a; x++)
{
 statements;
}
for (int y = 1; y <= b; y++)
{
 for (int z = 1; z <= c; z++)
 {
 statements;
 }
}
```
- (ii) How would the complexity change if all the three loops went to N instead of a, b and c?
- (e) Differentiate between a *constructor* and a *method* of a class. [2]

## Question 3

The following function **magicfun( )** is a part of some class. What will the function **magicfun( )** return, when the value of **n=7** and **n=10**, respectively? Show the dry run/working: [5]

```
int magicfun(int n)
{ if (n == 0)
 return 0;
 else
 return magicfun(n/2) * 10 + (n % 2);
}
```

## 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  $F(A, B, C, D) = \Sigma (2,3,4,5,6,7,8,10,11)$ .
- (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  $F(P, Q, R, S) = \pi(0,1,2,4,5,6,8,10)$ .
- (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) A school intends to select candidates for an Inter-School Essay Competition as per the criteria given below: [5]
- The student has participated in an earlier competition and is very creative.
- OR**
- The student is very creative and has excellent general awareness, but has not participated in any competition earlier.
- OR**
- The student has excellent general awareness and has won prize in an inter-house competition.

The inputs are:

| INPUTS |                                         |
|--------|-----------------------------------------|
| A      | participated in a competition earlier   |
| B      | is very creative                        |
| C      | won prize in an inter-house competition |
| D      | has excellent general awareness         |

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

Output : **X** [1 indicates yes, 0 indicates no for all cases]

Draw the truth table for the inputs and outputs given above and write the POS expression for  $X(A,B,C,D)$ .



- (b) State the application of a *Half Adder*. Draw the truth table and circuit diagram for a Half Adder. [3]
- (c) Convert the following Boolean expression into its canonical POS form: [2]  

$$F(A,B,C) = (B + C') \cdot (A' + B)$$

### Question 6

- (a) What is a *Multiplexer*? How is it different from a *decoder*? Draw the circuit diagram for a 8:1 Multiplexer. [5]
- (b) Prove the Boolean expression using Boolean laws. Also, mention the law used at each step. [3]

$$F = (x' + z) + [(y' + z) \cdot (x' + y)]' = 1$$

- (c) Define *maxterms* and *minterms*. Find the maxterm and minterm when: [2]  
 $P = 0, Q = 1, R = 1 \text{ and } S = 0$

### 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 class **Palin** has been defined to check whether a positive number is a *Palindrome* number or not. [10]

The number 'N' is palindrome if the original number and its reverse are same.

Some of the members of the class are given below:

|                                         |                                                                                                                                           |
|-----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Class name</b>                       | : <b>Palin</b>                                                                                                                            |
| <b>Data members/instance variables:</b> |                                                                                                                                           |
| num                                     | : integer to store the number                                                                                                             |
| revnum                                  | : integer to store the reverse of the number                                                                                              |
| <b>Methods/Member functions:</b>        |                                                                                                                                           |
| Palin( )                                | : constructor to initialize data members with legal initial values                                                                        |
| void accept( )                          | : to accept the number                                                                                                                    |
| int reverse(int y)                      | : reverses the parameterized argument 'y' and stores it in 'revnum' using <b>recursive technique</b>                                      |
| void check( )                           | : checks whether the number is a Palindrome by invoking the function <b>reverse( )</b> and display the result with an appropriate message |

Specify the class **Palin** giving the details of the **constructor( )**, **void accept( )**, **int reverse( 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 **Adder** has been defined to add any two accepted time.

[10]

Example: Time A - 6 hours 35 minutes

Time B - 7 hours 45 minutes

Their sum is - 14 hours 20 minutes ( where 60 minutes = 1 hour)

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

**Class name** : **Adder**

**Data member/instance variable:**

**a[ ]** : integer array to hold two elements (hours and minutes)

**Member functions/methods:**

**Adder( )** : constructor to assign 0 to the array elements

**void readtime( )** : to enter the elements of the array

**void addtime( Adder X, Adder Y )** : adds the time of the two parameterized objects X and Y and stores the sum in the current calling object

**void disptime( )** : displays the array elements with an appropriate message (i.e. hours = and minutes = )

Specify the class **Adder** giving details of the **constructor( )**, **void readtime( )**, **void addtime(Adder, Adder)** and **void disptime( )**. Define the **main( )** function to create objects and call the functions accordingly to enable the task.

### Question 9

A class **SwapSort** has been defined to perform string related operations on a word input. [10]

Some of the members of the class are as follows:

**Class name** : **SwapSort**

**Data members/instance variables:**

**wrd** : to store a word  
**len** : integer to store length of the word  
**swapwrd** : to store the swapped word  
**sortwrd** : to store the sorted word

**Member functions/methods:**

**SwapSort()** : default constructor to initialize data members with legal initial values  
**void readword()** : to accept a word in UPPER CASE  
**void swapchar()** : to interchange/swap the first and last characters of the word in '**wrd**' and stores the new word in '**swapwrd**'  
**void sortword()** : sorts the characters of the original word in alphabetical order and stores it in '**sortwrd**'  
**void display()** : displays the original word, swapped word and the sorted word

Specify the class **SwapSort**, giving the details of the **constructor()**, **void readword()**, **void swapchar()**, **void sortword()** 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 10

A *super class* **Product** has been defined to store the details of a product sold by a wholesaler to a retailer. Define a *sub class* **Sales** to compute the total amount paid by the retailer *with or without fine* along with *service tax*. [5]



Some of the members of both the classes are given below:

**Class name** : **Product**

**Data member/instance variable:**

name : stores the name of the product  
code : integer to store the product code  
amount : stores the total sale amount of the product (in decimals)

**Member functions/methods:**

Product(String n, int c, double p) : parameterized constructor to assign data members name=n, code=c and amount = p  
void show() : displays the details of the data members

**Class name:** **Sales**

**Data member/instance variable:**

day : stores number of days taken to pay the sale amount  
tax : to store the service tax (in decimals)  
totamt : to store the total amount (in decimals)

**Member functions/methods:**

Sales(...) : parameterized constructor to assign values to data members of both the classes  
void compute() : calculates the service tax @ 12.4% of the actual sale amount  
calculates the fine @ 2.5% of the actual sale amount **only** if the amount paid by the retailer to the wholesaler exceeds 30 days  
calculates the total amount paid by the retailer as (actual sale amount + service tax + fine)  
void show() : displays the data members of super class and the total amount

Assume that the super class **Product** has been defined. Using the **concept of inheritance**, specify the class **Sales** giving the details of the **constructor(...)**, **void compute( )** and **void show( )**.

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

### Question 11

Queue is an entity which can hold a maximum of 100 integers. The queue enables the user to add integers from the rear and remove integers from the front.

[5]

Define a class **Queue** with the following details:

**Class name** : **Queue**

#### **Data Members / instance**

**variables:**

**Que[ ]** : array to hold the integer elements

**size** : stores the size of the array

**front** : to point the index of the front

**rear** : to point the index of the rear

#### **Member functions:**

**Queue (int mm)** : constructor to initialize the data  
size = mm, front = 0, rear = 0

**void addele(int v )** : to add integer from the rear if possible  
else display the message "**Overflow**"

**int delele( )** : returns elements from front if present,  
otherwise displays the message  
"**Underflow**" and return -9999

**void display ( )** : displays the array elements

Specify the class **Queue** giving details of **ONLY** the functions **void addele(int)** and **int delele( )**. Assume that the other functions have been defined.

The main function and algorithm 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
{
 int num;
 Node next;
}
```

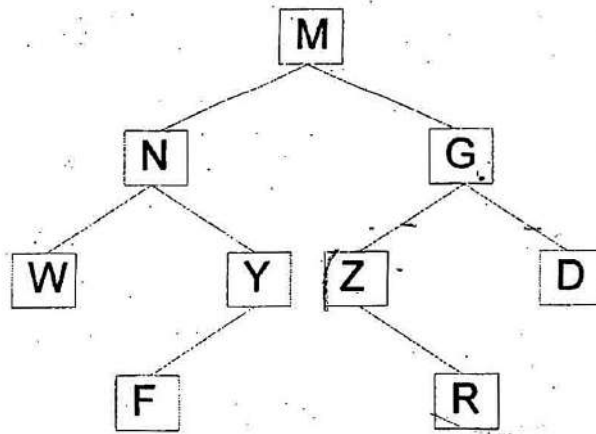
Write an *Algorithm* OR a *Method* to count the nodes that contain only odd integers from an existing linked list and returns the count.

The method declaration is as follows:

**int CountOdd( Node startPtr )**



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



- (i) Write the postorder traversal of the above tree structure. [1]
- (ii) State the level numbers of the nodes **N** and **R** if the root is at **0** (zero) level. [1]
- (iii) List the internal nodes of the right sub-tree. [1]

# COMPUTER SCIENCE

## Paper – 2

### (PRACTICAL)

(Maximum Marks: 30)

(Time allowed: Three hours)

(Candidates are allowed additional 15 minutes for only reading the paper.

They must NOT start writing during this time.)

---

**The total time to be spent on the Planning Session and the Examination Session is Three hours.**

Planning session: 90 minutes

Examination session : 90 minutes

**Note: Candidates are to be permitted to proceed to the Examination Session only after 90 minutes of the Planning session are over.**

---

*This paper consists of **three** problems from which candidates are required to attempt **any one** problem.*

Candidates are expected to do the following:

1. Write an algorithm for the selected problem. [3]  
(Algorithm should be expressed clearly using any standard scheme such as pseudo code or in steps which are simple enough to be obviously computable.)
2. Write a program in **JAVA** language. The program should follow the algorithm and should be logically and syntactically correct. [5]
3. Document the program using mnemonic names / comments, identifying and clearly describing the choice of data types and meaning of variables. [2]
4. Code / Type the program on the computer and get a printout ( hard copy ). Typically, this should be a program that compiles and runs correctly. [2]
5. Test run the program on the computer using the given sample data and get a printout of the output in the format specified in the problem. [5]
6. Viva-Voce on the Selected Problem. [3]

In addition to the above, the practical file of the candidate containing the practical work related to programming assignments done during the year is to be evaluated as follows:

- Programming assignments done throughout the year (by the teacher) [5]
- Programming assignments done throughout the year (by the Visiting Examiner) [5]

---

**This Paper consists of 5 printed pages and 1 blank page.**

1217-868B

© Copyright reserved.

**Turn over**

[www.guideforschool.com](http://www.guideforschool.com)

*Solve any one of the following Problems.*

### Question 1

A company manufactures packing cartons in four sizes, i.e. cartons to accommodate 6 boxes, 12 boxes, 24 boxes and 48 boxes. Design a program to accept the number of boxes to be packed (N) by the user (maximum up to 1000 boxes) and display the break-up of the cartons used in descending order of capacity (i.e. preference should be given to the highest capacity available, and if boxes left are less than 6, an extra carton of capacity 6 should be used.)

Test your program with the following data and some random data:

#### Example 1

**INPUT:** N = 726

**OUTPUT:**

|                         |                      |
|-------------------------|----------------------|
|                         | $48 \times 15 = 720$ |
|                         | $6 \times 1 = 6$     |
| Remaining boxes         | = 0                  |
| Total number of boxes   | = 726                |
| Total number of cartons | = 16                 |

#### Example 2

**INPUT:** N = 140

**OUTPUT:**

|                         |                    |
|-------------------------|--------------------|
|                         | $48 \times 2 = 96$ |
|                         | $24 \times 1 = 24$ |
|                         | $12 \times 1 = 12$ |
|                         | $6 \times 1 = 6$   |
| Remaining boxes         | $2 \times 1 = 2$   |
| Total number of boxes   | = 140              |
| Total number of cartons | = 6                |

#### Example 3

**INPUT:** N = 4296

**OUTPUT:** INVALID INPUT



## Question 2

The result of a quiz competition is to be prepared as follows:

The quiz has five questions with four multiple choices (A, B, C, D), with each question carrying 1 mark for the correct answer. Design a program to accept the number of participants N such that N must be greater than 3 and less than 11. Create a double dimensional array of size (N×5) to store the answers of each participant row-wise. Calculate the marks for each participant by matching the correct answer stored in a single dimensional array of size 5. Display the scores for each participant and also the participant(s) having the highest score.

**Example:** If the value of N = 4, then the array would be :

|               | Q.1 | Q.2 | Q.3 | Q.4 | Q.5 |
|---------------|-----|-----|-----|-----|-----|
| Participant 1 | A   | B   | B   | C   | A   |
| Participant 2 | D   | A   | D   | C   | B   |
| Participant 3 | A   | A   | B   | A   | C   |
| Participant 4 | D   | C   | C   | A   | B   |

Key to the question:

|   |   |   |   |   |
|---|---|---|---|---|
| D | C | C | A | B |
|---|---|---|---|---|

**Note:** Array entries are line fed (i.e. one entry per line)

Test your program for the following data and some random data:

### Example 1

**INPUT:**            N = 5  
Participant 1   D A B C C  
Participant 2   A A D C B  
Participant 3   B A C D B  
Participant 4   D A D C B  
Participant 5   B C A D D  
**Key :** B C D A A

**OUTPUT:**        **Scores:**  
Participant 1 = 0  
Participant 2 = 1  
Participant 3 = 1  
Participant 4 = 1  
Participant 5 = 2  
Highest score: Participant 5

### Example 2

**INPUT:** N = 4  
Participant 1 A C C B D  
Participant 2 B C A A C  
Participant 3 B C B A A  
Participant 4 C C D D B  
**Key :** A C D B B

**OUTPUT:** **Scores:**  
Participant 1 = 3  
Participant 2 = 1  
Participant 3 = 1  
Participant 4 = 3  
Highest score: Participant 1  
Participant 4

### Example 3

**INPUT:** N = 12  
**OUTPUT:** INPUT SIZE OUT OF RANGE.

### Question 3

**Caesar Cipher** is an encryption technique which is implemented as ROT13 ('rotate by 13 places'). It is a simple letter substitution cipher that replaces a letter with the letter 13 places after it in the alphabets, with the other characters remaining unchanged.

#### ROT13

|     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| A/a | B/b | C/c | D/d | E/e | F/f | G/g | H/h | I/i | J/j | K/k | L/l | M/m |
| ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   | ↕   |
| N/n | O/o | P/p | Q/q | R/r | S/s | T/t | U/u | V/v | W/w | X/x | Y/y | Z/z |

Write a program to accept a plain text of length L, where L must be greater than 3 and less than 100.

Encrypt the text if valid as per the Caesar Cipher.

Test your program with the sample data and some random data:

### Example 1

**INPUT:** Hello! How are you?  
**OUTPUT:** The cipher text is:  
Uryyb? Ubj ner lbh?

### **Example 2**

**INPUT:** Encryption helps to secure data.

**OUTPUT:** The cipher text is:

Rapelcgvba Urycf gb frpher gngn.

### **Example 3**

**INPUT:** You

**OUTPUT:** INVALID LENGTH

G T F S