ISC

COMPUTER SCIENCE

PAPER 1 (THEORY)

- **SOLUTION OF 2011**
- **COMMENTS OF COUNCIL EXAMINERS**
- **SUGGESTIONS FOR TEACHERS**

Dedicated to all my lovely students. May God help you always.

This small booklet contains solution of 2011 ISC Computer Science Paper 1 (Theory). The comments from the council examiners under solution of every question makes this a very handy guide for students to understand what the council expects as answer from the students.

I hope that the students will find this to be useful.

Md. Zeeshan Akhtar

1st February, 2015

BLANK PAGE

COMPUTER SCIENCE PAPER 1 (THEORY)

Question 1

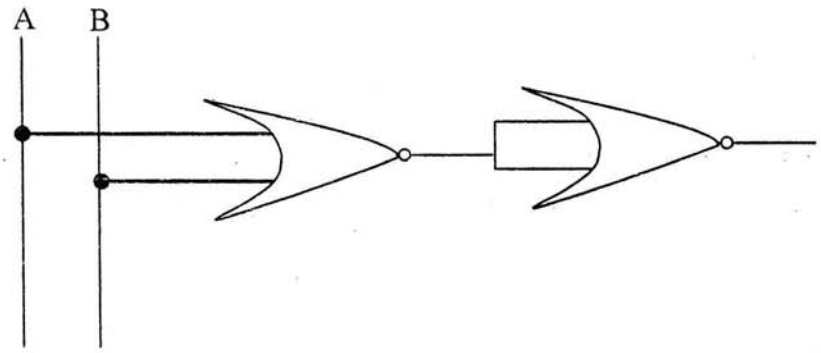
(a) State the two Absorption laws. Verify any one of them using truth table. **[2]**

(b) Reduce the following expression: **[2]**

$F(A,B,C) = \sum (0,1,2,3,4,5,6,7)$

Also find the complement of the reduced expression.

(c) Name the logic gate for the following circuit diagram and write its truth table. **[2]**



(d) Using truth table, verify whether the following is true or false: **[2]**

$$(p \Rightarrow q) = (\overline{q} \Rightarrow \overline{p})$$

(e) If A=1, B=0, C=1 and D=1 find its : **[2]**

(i) Maxterm

(ii) Minterm

Comments of Examiners

(a) Most of the candidates answered this question correctly. Some verified with the help of Boolean laws instead of truth table. Some used alternative laws like A+A'B = A+B and A.(A'+B) = A.B.

(b) Many candidates tried to solve this problem algebraically. Some have made two quads instead of one octal. Some of the candidates were confused as the reduced expression becomes 1. Some did not write the complement. Some wrote 'Cannot be reduced' / 'Not possible'.

(c) Some candidates named the gate as NOR gate as shown in the diagram, but the resultant gate was not written. Truth table was answered correctly by most of the candidates.

(d) This question was answered correctly by most of the candidates Most of the candidates proved with the help of laws instead of using truth table. Some were confused with the symbol (=>). Some e tried to solve this problem using expression p'+ q to get the output.

Suggestions for teachers

- Laws and theorems must be explained and proved by the help of truth table and also by using laws.
- Tautology / Contradiction conditions must be explained to candidates and show how one expression is reduced to 1 or 0 respectively.
- Candidates should be told to answer the question as required. Drawing of logic gates using universal gates must be given more practice
- Teachers should give circuits to the students and also tell them to find the output.

www.javaforschool.com

(e) Candidates either scored full credit or not scored at all in this question. Some interchanged the maxterms with the minterms and vice-versa.

- Propositional logic and different types of operators / symbols and their uses should be explained using well formed formulae.
- Generation of maxterms and minterms either from truth table or random selection must be explained in detail. The difference between Maxterms and Minterms must be explained thoroughly.

MARKING SCHEME

Question 1.

(a) Absorption law states that :

$A + A.B = A$

$A.(A+B) = A$

A	B	A+B	A.(A+B)
0	0	0	0
0	1	1	0
1	0	1	1
1	1	1	1

(b)

	B'C'	B'C	BC	BC'
A'	0 **1**	1 **1**	3 **1**	2 **1**
A	4 **1**	5 **1**	7 **1**	6 **1**

Output = 1 ; complement = 0

(c) Logic gate is OR gate (A + B)

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	1

(d)

p	q	$p \Rightarrow q$	$\bar{q}$	$\bar{p}$	$\bar{q} \Rightarrow \bar{p}$
0	0	1	1	1	1
0	1	1	0	1	1
1	0	0	1	0	0
1	1	1	0	0	1

www.javaforschool.com

(e) A B C D

1 0 1 1

Maxterm = A' + B + C' + D'

Minterm = AB'CD

Question 2

(a) How can we override a method in inheritance? **[2]**

(b) A square matrix A[m x m] is stored in the memory with each element requiring 2 bytes of storage. If the base address A[1][1] is 1098 and the address at A [4][5] is 1144, determine the order of the matrix A[m x m] when the matrix is stored **Column Major** wise. **[2]**

(c) What is Big O notation? **[2]**

(d) What is an exception? **[2]**

(e) Convert the following infix expression to its postfix form: **[2]**

a + b * c – d / e

Comments of Examiners

(a) Most of the candidates used the word 'Super' to override a method in inheritance. Some wrote the correct answer. Instead of writing how can we override, some candidates wrote how to prevent overriding.

(b) Candidates having the knowledge of the formula to find the address of a cell in an array could solve the question and get the correct answer. Others have answered vaguely.

(c) Most of the candidates answered this question well. Some wrote vague definition. Some gave only the examples without giving the definition.

(d) Most of the candidates answered this question correctly. Some gave the different types of exceptions. Some mentioned only the examples. Some students were confused with exceptions as syntax errors.

(e) Most candidates were able to solve this problem correctly. Some candidates wrote the correct answer without showing the working. Some applied the postfix correctly, but could not derive the final answer.

Suggestions for teachers

- More practice should be given in overriding methods in inheritance with the help of examples.
- Practice should be given in understanding the formula using row major and column major and to make the subject of formula that is required in the question.
- Definition should be given and explained with examples. The use and importance of Big 'O' notation must be explained.
- More practice of exception and exceptional handling should be given to the candidates. Different types of exceptions should also be explained in details.
- Examples need to be practiced with conversion of Infix to Postfix notation, the order of precedence and also the Polish string method.

www.javaforschool.com

MARKING SCHEME

Question 2.

(a) Overriding occurs whenever a subclass has a method with same signature as a method in one of its superclasses. As a result the method in the derived class overrides the method in superclass.

(b) **Column major address** formula : $A[i][j] = BA + W\ [(j - l_c) * row + (i - l_r)]$

$BA = 1098,\ l_r = 1\ ,\ l_c = 1\ ,\ W = 2\ ,\ i = 4\ ,\ j = 5\ ,\ A[i][j] = 1144$

$1144 = 1098 + 2\ [(5-1)m + 4-1]$

$1144 = 1098 + 8m + 6$

$8m = 40\ ;\ m = 40/8\ ;$ **m= 5 Ans**

(c) **Big O Notation** is the measurement of the complexity of an algorithm with respect to time factor and space (memory) factor.

(d) An exception is an unexpected or anomalous situation or error during a program execution.

(e) a + b * c – d / e

a + bc* - de/

a b c * + d e / -

Question 3

(a) The following is a part of some class. What will be the output of the function **mymethod( )** when the value of counter is equal to 3? Show the dry run / working. **[5]**

```
void mymethod(int counter)
 {
   if ( counter = = 0 )
       System.out.println(" ");
   else
     {
       System.out.println("Hello"+counter);
       mymethod(– –counter ) ;
       System.out.println("  "+counter);
     }
 }
```


www.javaforschool.com

(b) The following function is a part of some class which computes and returns the greatest common divisor of any two numbers. There are some places in the code marked by **?1?, ?2?, ?3?, ?4?** and **?5?** which must be replaced by statement/expression so that the function works correctly.

```
int gcd(int a, int b)
 {
      int r;
      while(?1?)
       {
          r=?2?;
          b=?3?;
          a=?4?;
       }
      if(a==0)
          return ?5?;
      else
          return -1;
 }
```

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

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

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

(iv) What is the expression or statement at **?4?** **[1]**

(v) What is the expression or statement at **?5?** **[1]**

Comments of Examiners

(a) Most of the candidates answered this question correctly. Some candidates were not clear with the concept of recursive technique. Some did not show the working / dry run. Some were confused as the counter itself was decrementing. The second part of the output was missing in some cases. No working was shown in some cases.

(b) Some candidates were not clear with Greatest Common Divisor program without using recursion technique and hence could not answer all the parts of the question correctly. Since the while loop did not have values, some candidates could not answer the question correctly. Part (i) Answered correctly by most of the candidates, (ii) a%b was written instead of b%a in some cases (iii) vague answers were given (iv) answer of part(iii) was written in some cases and (v) most have written zero as a guess.

Suggestions for teachers

- Practice should be given more on program using recursive technique.
- Teachers are expected to show the dry run/ working of program using the concept of stack.
- Various techniques relating to problem solving should be taught and their differences along with their technique must be explained.

www.javaforschool.com

MARKING SCHEME

Question 3.

(a) Output:

```
Hello3
Hello2
Hello1

0
1
2
```

(b) (i) a > 0 or a != 0

(ii) r = b % a

(iii) b = a

(iv) a = r

(v) b

PART – II

*Answer **seven** questions in this part, choosing **three** questions from Section A, **two** from Section B and **two** from Section C.*

SECTION - A

*Answer any **three** questions.*

Question 4

(a) State the principle of Duality. Give the dual of the following : [3]

$(A'.B)+(C.1)=(A'+C).(B+C)$

(b) Reduce the Boolean expressions to their simplest forms: [4]

(i) $\{(C.D)'+A\} + A + C.D + A.B$

(ii) $A.\{B+C(A.B+A.C)'\}$

(c) Verify using a truth table if : [3]

$(A \odot B \odot C)' = A \oplus B \oplus C$

Comments of Examiners

(a) Most of the candidates answered this question correctly. Principle of Duality was explained in simple words rather than giving the correct definition. In some cases the dual was incorrect. Some candidates found the dual after simplifying the equation.

(b) Answered well by most of the candidates. Some have used very lengthy method to reduce the expression.

Suggestions for teachers

- More practice needs to be given to find the dual of an equation. The definition should be explained with examples.
- Proper use of Laws / theorems in reducing Boolean expression should be practiced more.

www.javaforschool.com

(c) Most of the candidates answered this question directly that XOR is a complement of XNOR. But it is true only in even number of inputs and not in odd numbers as inputs.

- Apart from the textbooks, the candidates may be given notes that are gathered from the net or any other reliable source. (i.e. text books should not be the only medium).

MARKING SCHEME

Question 4.

(a) The principle of duality states that from every Boolean relation another Boolean relation can be derived by changing each OR (+) sign to an AND (.) sign, changing each AND (.) sign to an OR (+) sign and replacing each 1 by 0 and each 0 by 1 (complements remain unchanged). The new derived relation is known as the dual of the original relation.

Dual = (A` + B). (C + 0) = (A'.C) + (B.C)

(b) (i) [(C.D)' + A] + A + C.D + A.B

[C' + D' + A] + A.(1+B) + C.D

C' + D' + A + A.1 + C.D

C' + C.D + D' + A + A

C` + D + D` + A

1 + C' + A = 1 + A= 1

(ii) A.[B+C.(A.B + A.C)']

A.[B+C.((A.B)' .(A.C)']

A.[B+C.(A` + B`).(A' + C')]

A.[B+C.(A' + A`.C` + A'B' + B'C')]

A.[B+C.(A'+ B'C')]

A.B

(c)

A	B	C	A⊖B	A⊖B⊖C	LHS	A⊕B	A⊕B⊕C
0	0	0	1	0	1	0	0
0	0	1	1	1	0	0	1
0	1	0	0	1	0	1	1
0	1	1	0	0	1	1	0
1	0	0	0	1	0	1	1
1	0	1	0	0	1	1	0
1	1	0	1	0	1	0	0
1	1	1	1	1	0	0	1

www.javaforschool.com

Question 5

(a) Given $F(P, Q, R, S) = \pi(2, 3, 6, 7, 9, 11, 12, 13, 14, 15)$ [5]

Reduce the above expression by using ***four*** variable Karnaugh's Map. Draw the logic gate diagram of the reduced expression using NOR gate only.

(b) Given $F(A, B, C, D) = \bar{A}\,\bar{B}\,\bar{C}\,\bar{D} + \bar{A}\,\bar{B}\,\bar{C}\,D + A\,\bar{B}\,\bar{C}\,\bar{D} + A\,\bar{B}\,\bar{C}\,D + \bar{A}\,B\,\bar{C}\,\bar{D} + \bar{A}\,B\,C\,\bar{D}$. [5]

Reduce the above expression by using ***four*** variable Karnaugh's Map. Draw the logic gate diagram of the reduced expression using NAND gate only.

Comments of Examiners

(a) Most candidates answered this question correctly. Some candidates committed errors in place value and putting variables in K-Map. In some cases the groups were reduced by laws. Some candidates drew the K-Map incorrectly. Many candidates included the redundant group in the final expression. Logic circuit were not drawn correctly, instead using NOR gates, basic gates were used.

(b) Most candidates fared well in this part. Some candidates were not able to draw the K-Map for the POS expression correctly . For a number of candidates the "Map rolling" concept was not very clear. Some have converted the canonical form to cardinal form and then reduced it.

Suggestions for teachers

- Emphasize on arranging the variables in proper order and the importance of cell values corresponding with the variables. Explain clearly how the group are framed and reduced. Redundant groups are not to be included in the final reduced expression. Use of gray code for labeling K-Map should be explained clearly.
- Students should be instructed to reduce POS and SOP expressions using K-Map simultaneously. Students should be told not to include the redundant group in the final expression.

www.javaforschool.com

MARKING SCHEME

Question 5.

(a) $F(P,Q,R,S) = \pi(2, 3, 6, 7, 9, 11, 12, 13, 14, 15)$

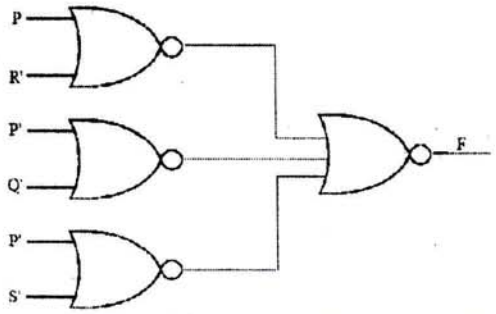
	R+S	R+S'	R'+S'	R'+S
P+Q	0 1	1 1	3 **0**	2 **0**
P+Q'	4 1	5 1	7 **0**	6 **0**
P'+Q'	12 **0**	13 **0**	15 **0**	14 **0**
P'+Q	8 1	9 **0**	11 **0**	10 1

There are three quads :

Quad 1 ($M_2 M_3 M_6 M_7$) = P + R' ; Quad 2 ($M_{12} M_{13} M_{14} M_{15}$) = P' + Q'

Quad 3 ($M_9 M_{11} M_{13} M_{15}$) = P' + S'

F(P,Q,R,S) = (P + R') . (P' + Q') . (P' + S')



www.javaforschool.com

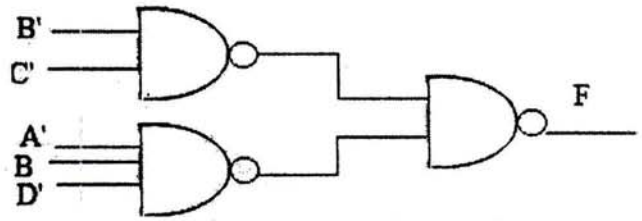
(b) **F(A,B,C,D) = A'B'C'D' + A'B'C'D + AB'C'D' + AB'C'D + A'BC'D' + A'BCD'**

	C'D'	C'D	CD	CD'
A'B'	0 **1**	1 **1**	3 0	2 0
A'B	4 **1**	5 0	7 0	6 **1**
AB	12 0	13 0	15 0	14 0
AB'	8 **1**	9 **1**	11 0	10 0

There is one quad and one pair :

Quad $(m_0 + m_1 + m_8 + m_9)$ = B'C' ; Pair $(m_4 + m_6)$ = A'BD'

F(A,B,C,D) = B'C' + A'BD'



Question 6

(a) Show with the help of a logic diagram how a NAND gate is equivalent to an OR gate. [3]

(b) Verify if the following is valid : [3]

$(a \Rightarrow b) \wedge (a \Rightarrow c) = a \Rightarrow (b \wedge c)$

(c) What is a Decoder? Draw the truth table and logic circuit diagram for a 2 to 4 Decoder. [4]

Comments of Examiners

(a) Most of the candidates answered this question correctly. Some have drawn the simple OR gate without using the NAND gates. Some have used expression to show that a NAND gate is equivalent to an OR gate.

(b) Candidates were confused with the symbols (Λ) and (=>). Only some candidates could answer this part correctly. Others have only drawn the input combinations of the truth table. Some students used OR for (Λ) symbol instead of (.) or AND.

Suggestions for teachers

- More practice should be given in proving the universality of NAND and NOR gates as AND, OR and NOT gates.
- Practice on propositional logic and the use of operators / symbols in proving a well formed formulae using truth table must be emphasized.

www.javaforschool.com

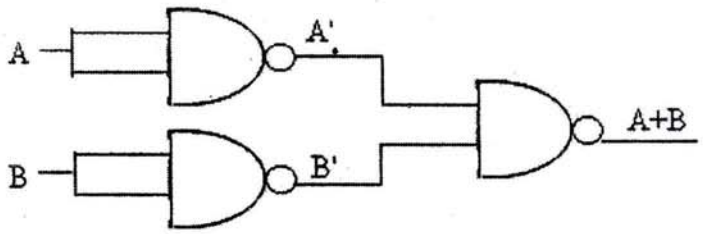
(c) Many candidates answered this question correctly. Some candidates used the OR gate instead of AND gate in drawing decoder's logic circuit. In some cases even the truth table was incorrect. Some confused the diagram of encoder with decoder.

- Practice should be given to draw the encoder and decoder and their differences should be explained along with the truth table. Use of proper gate and the flow lines should be highlighted.

MARKING SCHEME

Question 6.

(a)



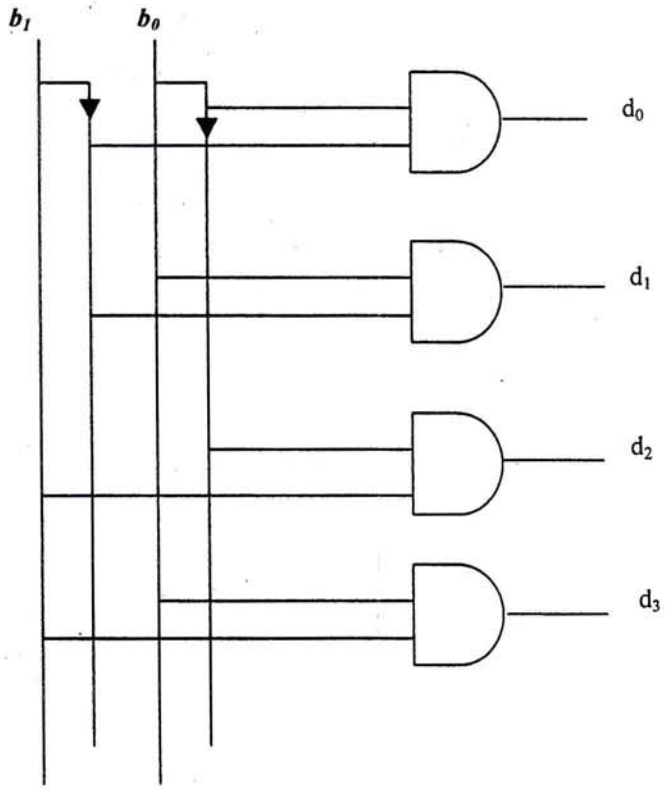
(b)

a	b	c	a => b	a => c	L.H.S	$(b \cap c)$	$a => (b \cap c)$
0	0	0	1	1	1	0	1
0	0	1	1	1	1	0	1
0	1	0	1	1	1	0	1
0	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0
1	0	1	0	1	0	0	0
1	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1

(c) A decoder is a combinational circuit which inputs 'n' lines and outputs 2^n or fewer lines and converts a binary number to its equivalent denary form.

Inputs		Outputs			
b_1	b_0	d_0	d_1	d_2	d_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

www.javaforschool.com



Question 7

(a) What is a Full Adder? Draw the truth table for a Full Adder. Also derive SOP expression for the Full Adder and draw its logic circuit. **[4]**

(b) State how a Decoder is different from a Multiplexer. Also state one use of each. **[3]**

(c) Convert the following cardinal expression into its canonical form and reduce it using Boolean laws: **[3]**

$$F(L,M,O,P) = \pi\,(0,2,8,10)$$

Comments of Examiners

(a) Most of the candidates answered this question correctly. Some had written the expression in POS form. The logic diagram and truth table were also answered correctly by most of the candidates. Some could not write the SOP expression for SUM, but have written the SOP expression for CARRY correctly.

(b) The term 'combinational circuit' was not used in the definition by many candidates. Some were able to give suitable examples of their uses / applications.

(c) Most of the candidates answered this question correctly. Some could not convert the cardinal expression into its canonical form. Some used the K-Map to reduce the expression. In some cases mistakes were made in reducing the expression.

Suggestions for teachers

- Practice should be given in drawing the full adder and half adder along with its expression, truth table and logic circuit / diagram.
- The students should be encouraged to write the correct definition and also differentiate between decoders and multiplexers along with their uses.
- Cardinal and canonical form of expression must be taught in detail. The inter changing of the two types of expression must be given more practice.

www.javaforschool.com

MARKING SCHEME

Question 7.

(a) Full adder is a combinational circuit which adds 3 input binary bits and outputs 2 binary bits.

Truth table for Full Adders :

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Expression of full adder :

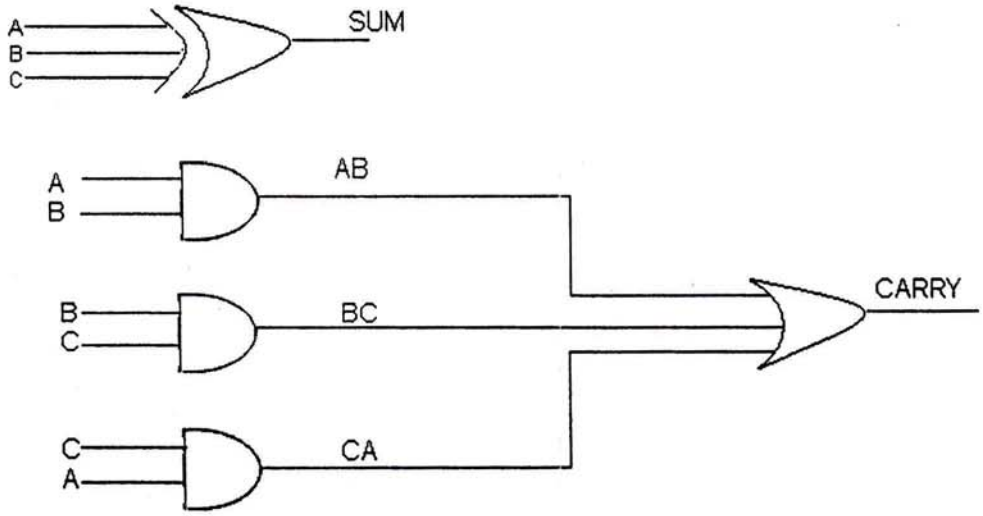
For Sum : A'B'C + A'BC' + AB'C' + ABC

$A \oplus B \oplus C$

For carry : A'BC + AB'C + ABC' + ABC

: AB + BC + AC

Logic gate diagram :



(b) **Decoder :** It is a combinational circuit which inputs 'n' lines and outputs 2^n or fewer lines. It is used for converting binary numbers to octal, decimal and hexadecimal numbers.

Multiplexer : It is a combinational circuit which inputs parallel data lines and outputs the serial data line. It is used for data transmissions.

(c) $F(L,M,O,P) = \pi\,(0,2,8,10)$

(L+M+O+P) . (L+M+O'+P) . (L'+M+O+P) . (L'+M+O'+P)

(L+M+P) . (O'+O) . (L'+M+P) . (O+O')

= (M+P) . (L+L')

= **M+P**

www.javaforschool.com

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 8

Input a sentence from the user and count the number of times, the words "an" and "and" are present in the sentence. Design a class **Frequency** using the description given below: **[10]**

Class name	:	**Frequency**
Data Members / variables:		
text	:	stores the sentence
countand	:	to store the frequency of the word "and"
countan	:	to store the frequency of the word "an"
len	:	stores the length of the string
Member functions/methods:		
Frequency( )	:	constructor to initialize the instance variables
void accept(String n)	:	to assign n to text, where the value of the parameter n should be in lower case.
void checkandfreq( )	:	to count the frequency of "and"
void checkanfreq( )	:	to count the frequency of "an"
void display( )	:	to display the number of "and" and "an" with appropriate messages.

Specify the class **Frequency** giving details of the **constructor( ), void accept(String), void checkandfreq( ), void checkanfreq( )** and **void display( )**. Also define the **main( )** function to create an object and call methods accordingly to enable the task.

Comments of Examiners

Most of the candidates answered this question correctly and have scored full credit. Some attempted using string tokenizer. Some of the candidates found the frequency of 'and' and 'an' not as a separate word, but as a occurrence in the sentence.
Declaration of variables was done inside the constructor. In accept( ) function some students had taken the inputs using readline( ) instead of parameter. The other functions were well answered including the constructor. The main function was not attempted by some of the candidates.

Suggestion for teachers

- Knowledge of constructors should be given more practice. Use of string tokenizer should also be practiced as it simplifies writing of the string program to a large extent.

www.javaforschool.com

MARKING SCHEME

Question 8.

```
import java.io.*;
public class Frequency
{    String text;
     int countand,countan,len;
     public Frequency()
       {  text=null;
          countand=0;
          countan=0;
       }
 public void accept(String n)
 {     text=n;
       len=text.length();
 }
 public void checkandfreq()
 {     int p=0;char c;String b;
       for(int i=0;i<len;i++)
         {    c=text.charAt(i);
              if(c==' '||c=='.')
                 {   b=text.substring(p,i);
                     if(b.equals("and"))
                       countand++;
                     p=i+1;
                 }
         }
  }
  public void checkanfreq()
  {      int p=0;char c;String b;
        for(int i=0;i<len;i++)
           {     c=text.charAt(i);
                 if(c==' '||c=='.')
                   {    b=text.substring(p,i);
                        if(b.equals("an"))
                          countan++;
                        p=i+1;
                    }
           }
  }
  public void display()
  {       System.out.println("Number of and's =" + countand);
          System.out.println("Number of an's =" + countan);
  }
  public static void main()throws IOException
   {     BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
         System.out.println("Enter a sentence in Lower case");
         String n= br.readLine();
         Frequency x= new Frequency();
         x.accept(n);
         x.checkandfreq();
         x.checkanfreq();
         x.display();
    }
}
```


www.javaforschool.com

Question 9

A class **DeciOct** has been defined to convert a decimal number into its equivalent octal number. Some of the members of the class are given below:

Class name	:	**DeciOct**
Data Members / instance variables:		
n	:	stores the decimal number
oct	:	stores the octal equivalent number
Member functions:		
DeciOct( )	:	constructor to initialize the data members n=0, oct = 0.
void getnum(int nn)	:	assign nn to n
void deci_oct( )	:	calculates the octal equivalent of 'n' and stores it in oct using the **recursive technique**
void show( )	:	displays the decimal number 'n', calls the function deci_oct( ) and displays its octal equivalent.

(a) Specify the class **DeciOct,** giving details of the **constructor( ), void getnum(int), void deci_oct( )** and **void show( ).** Also define a **main( )** function to create an object and call the functions accordingly to enable the task. [8]

(b) State any two disadvantages of using recursion. [2]

Comments of Examiners

(a) The concept of recursion was not clear in some cases. Checking the base case in a recursive function was not properly done. Some solved the program directly without using the recursive technique. Students have written nn = n instead of n = nn.

(b) Some candidates have overlooked this part in spite of answering the part(a) correctly. Most of the candidates answered this part of the question correctly.

Suggestions for teachers

- More practice should be given to solve programs using recursive techniques. Much attention should be paid by the teachers towards recursion and its techniques with examples. Knowledge of base case and recursive case should be given to the students for every program using recursive technique.
- Advantages and disadvantages of recursion and recursive technique should be given and problems related to the same be practiced.
- Memory allotment should be demonstrated using memory blocks (i.e. each recursive call creates a new memory block and closes using LIFO technique)

www.javaforschool.com

MARKING SCHEME

Question 9.

(a)

```
import java.io.*;
public class DeciOct
{
    int n,oct;
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
  public DeciOct()
  {
     n=0;oct=0;
  }
  public void getnum(int  nn)
  {
     n=nn;
  }
  public void deci_oct()
  {
      int c;
      if(n!=0)
       {
          c=n%8;
          n=n/8;
          deci_oct();
          oct=(oct*10)+c;
       }
  }
  public void show()
  {
        System.out.println("Decimal number="+n);
        deci_oct();
        System.out.println("Octal equivalent = "+oct);
  }
  public static void main()throws IOException
  {     System.out.println("Enter a decimal number");
        int v =Integer.parseInt(br.readLine());
        DeciOct x=new DeciOct();
        x.getnum(v);
        x.show();
  }
}
```

(b) Slow process , more memory

www.javaforschool.com

Question 10

You are given a sequence of N integers, which are called as pseudo arithmetic sequences (sequences that are in arithmetic progression). **[10]**

Sequence of N integers : 2, 5, 6, 8, 9, 12

we observe that 2 + 12 = 5 + 9 = 6 + 8 = 14.

The sum of the above sequence can be calculated as 14 x 3 = 42.

For sequence containing an odd number of elements the rule is to double the middle element, for example 2, 5, 7, 9, 12

= 2 + 12 = 5 + 9 = 7 + 7 = 14..

14 × 3 = 42 [middle element = 7]

A class **Pseudoarithmetic** determines whether a given sequence is a pseudo-arithmetic sequence.

The details of the class are given below:

Class name	:	**Pseudoarithmetic**
Data Members / instance variables:		
n	:	to store the size of the sequence
a [ ]	:	integer array to store the sequence of numbers
ans, flag	:	store the status
sum	:	store the sum of sequence of numbers
r	:	store the sum of the two numbers
Member functions:		
Pseudoarithmetic( )	:	default constructor
void accept(int nn)	:	to assign nn to n and to create an integer array. Fill in the elements of the array.
boolean check( )	:	return true if the sequence is a pseudo-arithmetic sequence otherwise returns false

Specify the class **Pseudoarithmetic**, giving the details of the **constructor( ), void accept(int)** and **boolean check( ).** Also define a **main( )** function to create an object and call the member functions accordingly to enable the task.

Comments of Examiners

Some of the candidates failed to understand the use of the variables ans and flag for checking. The loop and solving was correct in most of the cases . The function boolean check( ) was not attempted by some of the candidates . Wrong logic was used to calculate the sum. Students used simple loop to add all the elements to find the sum. The other functions were well answered including the constructor. The main function was not attempted by some candidates.

Suggestion for teachers

- Knowledge of boolean data type (i.e. true or false) must be explained in details. More practice should be given in writing the main function with each and every program.

www.javaforschool.com

MARKING SCHEME

Question 10.

```
import java.io.*;
public class Pseudoarithmetic
{
      static BufferedReader br= new BufferedReader(new InputStreamReader(System.in));
      int n,sum,r;
      boolean ans,flag;
      int a[];
   public Pseudoarithmetic()
   {  ans=true;
      flag=true;
   }
   public void accept(int nn)throws IOException
   {  n=nn;
      a=new int[n];
      System.out.println("Enter" + n + "elements");
      for(int i=0;i<n;i++)
         a[i]=Integer.parseInt(br.readLine());
   }
   public boolean check()
   {   r=a[0]+a[n-1];
       if(n%2==0)
        {   for(int i=0;i<n/2;i++)
              {   if((a[i]+a[n-1-i])!=r)
                  flag=false;
              }
        }
       else
        {   for(int j=0;j<=n/2;j++)
              {    if ((a[j] + a[n-1-j])!=r)
                   ans=false;
              }
        }
       if(flag==true)
          sum=n/2 * r;
       if(ans=true)
         sum=(n/2 + 1)*r;
       if(ans==false || flag==false)
         return false;
       else
         return true;
    }
    public static void main() throws IOException
    {   System.out.println("enter size of sequence");
        int nn=Integer.parseInt(br.readLine());
        Pseudoarithmetic x=new Pseudoarithmetic();
        x.accept(nn);
        System.out.println(x.check());
    }
}
```

SECTION – C

Answer any two questions.

Each Program / Algorithm should be written in such a way that it clearly depicts the logic of the problem step wise. This can also be achieved by using pseudo codes.

(Flowcharts are **not** required).

The programs must be written in Java.

The Algorithm must be written in general/standard form.

Question 11

A super class **Record** has been defined to store the names and ranks of 50 students. Define a sub-class **Rank** to find the highest rank along with the name. The details of both classes are given below: **[10]**

Class name	:	**Record**
Data Members / instance variables:		
name[ ]	:	to store the names of students
rnk[ ]	:	to store the ranks of students
Member functions:		
Record( )	:	constructor to initialize data members
void readvalues( )	:	to store the names and ranks
void display( )	:	displays the names and the corresponding ranks
Class name	:	**Rank**
Data members /instance variables:		
index	:	integer to store the index of the topmost rank
Member functions :		
Rank( )		constructor to invoke the base class constructor and to initialize index = 0
void highest( )		finds the index/location of the topmost rank and stores it in index without sorting the array.
void display( )		displays the names and ranks along with the name having the topmost rank.

Specify the class **Record** giving details of the **constructor( )**, **void readvalues( )** and **void display( )**. Using the **concept of inheritance**, specify the class **Rank** giving details of **constructor( )**, **void highest( )** and **void display( )**.

THE MAIN( ) FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

Comments of Examiners

Concept of Inheritance was not clear to some candidates. Constructor with inheritance was not answered correctly. Accessing the members of the super class by the derived class was not clear to some of the candidates. Some candidates declared the base class data members as private. String data members were not declared properly by some students. Use of 'extend' keyword in derived class was missing. Different methods were used to find the top most rank. Some of the candidates wrote both the program and algorithm in spite of being mentioned not to in the question paper. The rest of the functions were well answered.

Suggestions for teachers

- Practice should be given on inheritance to the students. Use of constructor using the base class member should be made clear. The students should be explained the different visibility modes and their accessing capability. Knowledge of calling the member function from the super class to the derived class must be made clear.

MARKING SCHEME

Question 11.

```
 import java.io.*;
public class Record
{
      BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
      protected String name[]=new String[5];
      protected int rnk[]=new int[5];
  public Record()
  {
      for(int i=0;i<5;i++)
       {
          name[i]=null;
          rnk[i]=0;
       }
  }
  public void readvalue()throws IOException
   {    System.out.println("enter names & ranks");
        for(int i=0;i<5;i++)
          {   name[i]=br.readLine();
              rnk[i]=Integer.parseInt(br.readLine());
           }
   }
  public void display()
   {          for(int i=0;i<5;i++)
              System.out.println(name[i] + " , " + rnk[i]);
    }
}
import java.io.*;
public class Rank extends Record
{       BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
        int index;
    public Rank()
    {
        super();
        index=0;
    }
```


www.javaforschool.com

```
public void highest()
{
    for(int i=1;i<5;i++)
        {   if(rnk[index]>rnk[i])
            index=i;
        }
}
public void display() throws IOException
{   readvalue();
    highest();
    super.display();
    System.out.println("Name of the highest rank holder : " + name[index]);
}
}
```

Question 12

Stack is a kind of data structure which can store elements with the restriction that an element can be added or removed from the top only.

The details of the class **Stack** is given below:

Class name	:	**Stack**
Data Members / instance variables:		
st[ ]	:	the array to hold the names
size	:	the maximum capacity of the string array
top	:	the index of the topmost element of the stack
ctr	:	to count the number of elements of the stack
Member functions:		
Stack( )	:	default constructor
Stack(int cap)	:	constructor to initialize size = cap and top = –1
void pushname(String n)	:	to push a name into the stack. If the stack is full, display the message "OVERFLOW"
String popname( )	:	removes a name from the top of the stack and returns it. If the stack is empty, display the message "UNDERFLOW"
void display( )	:	Display the elements of the stack.

(a) Specify class **Stack** giving details of the **constructors( ), void pushname (String n), String popname( )** and **void display( ).** [8]

THE MAIN( ) FUNCTION AND ALGORITHM NEED NOT BE WRITTEN.

(b) Under what Principle does the above entity work? [2]

www.javaforschool.com

Comments of Examiners

(a) Some candidates had problem in declaring and initializing the array. The constructors were not properly declared and the array was not created inside the constructor in some cases. The functions pushname( ) and popname( ) were not answered properly. Some of the candidates have written both the program and the algorithm. Some were confused as the data member 'ctr' was not required in the entire program. The rest of the functions were well answered.

(b) Well answered by most of the candidates. Some have given vague answers.

Suggestions for teachers

- More practice should be given in data structure programs like the stacks, queues, dequeues etc. Working must be shown as to how the stack or a queue performs (examples can be supportive).
- The concept of LIFO and FIFO must be explained to the students with examples related to the real world.
- Difference between stack and queue must be explained clearly.

MARKING SCHEME

Question 12.

(a)

```
public class Stack
{
    String st[]=new String[50];
    int size,top,ctr;
    public  Stack( ) { }
  public  Stack(int cap)
  {
     size=cap;
     top=-1;
     st=new String[size];
  }
  public void pushname(String n)
  {
     if (top<size-1)
         st[++top]=n;
     else
         System.out.println(" OVERFLOW ");
  }
  public String  popname( )
  {   String  v;
      if(top>=0)
       {
           v=st[top--];
           return v;
       }
      else
      {
          System.out.println(" UNDERFLOW ");
          return " ";
      }
   }
  public void display( )
```


www.javaforschool.com

```
        {
            for(int i=top;i>=0;i--)
                System.out.println(st[i]);
        }
    }
```

(b) It works on the principle of **LIFO** (Last In First Out) operations.

Question 13

(a) A linked list is formed from the objects of the class, [4]

```
class Node
 {
   int info ;
   Node link ;
 }
```

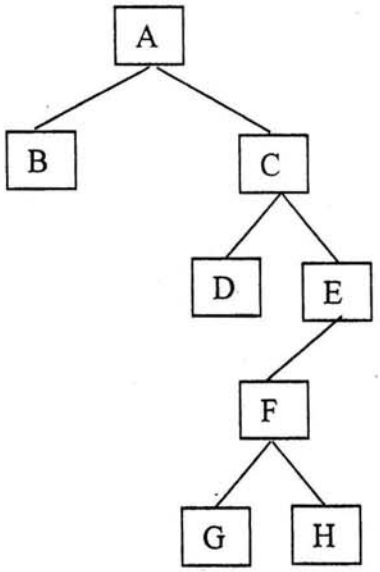
Write an algorithm **OR** a Method for deleting a node from a linked list.

The method declaration is given below :

void deletenode (Node start)

(b) Distinguish between worst-case and best case complexity of an algorithm. [2]

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



(i) Write the Postorder tree traversal [1]

(ii) Name the Leaves of the tree [1]

(iii) Height of the tree [1]

(iv) Root of the tree [1]

www.javaforschool.com

Comments of Examiners

(a) Some have written the algorithm for all the three cases where as others mentioned only one. Some candidates wrote the algorithm while some wrote in program/code form. Concept of linked list was not clear to the students.

(b) Most of the candidates answered this question correctly. Some of the candidates did not write the difference between best case and the worst case complexity. They have simply given definition of each separately.

(c) Some candidates were confused with preorder, inorder and postorder tree traversal method. Most of the candidates answered this part of the question correctly. (ii) Some candidates included the other nodes also. This part of the question was well answered by most. (iii) and (iv) was answered correctly by almost all the candidates.

Suggestions for teachers

- More programs / algorithms should be practiced with linked list and binary tree data structure. Difference between a linked list and an array should be explained.
- Concept of pointers with reference to pointing to a location and how to change the reference when element is inserted or removed must be explained diagrammatically.
- Definition of complexities / big 'O' and the three cases of complexities must be explained in detail along with examples.
- Explain binary tree with the different parts like root, nodes (internal and external), height, depth, level, size, tree traversal (preorder, inorder and postorder) etc.

MARKING SCHEME

Question 13.

(a) **Algorithm to delete a node in a linked list.**

Steps :

1 - Start

2 - Accept info for the node to be deleted

3 - if first node is the desired one then

Set temporary pointer to first node

Move the pointer of the first node to the next node

Set temporary pointer to null

Stop else

4 - move temporary pointer to the next node

5 - if the desired node is found then

Set the pointer of the previous node to the next node of the node to be deleted

Set the temporary pointer node to null

Stop else

6 - move temporary pointer to the next node

7 - Repeat step 5 & 6 until the temporary pointer reaches null

8 - End

Method to delete a node in a linked list

```
void deletenode ( Node start )
        {   System.out.println("enter info for the node to be deleted");
        int  n=Integer.parseInt(obj.readLine());
          Node a = new Node(start)
        if(start.info==n)
            {  a=start;
            start = start.link;
            a.link=null;   }
         else
            {     a=a.link;
              Node b=new Node(start);
                 while(a!=null)
                 {   if(a.info==n)
                      b.link=a.link;
                      a.link=null; break;
                     }
               b=a;
              a=a.link;
             } }
```

(b) **Best case complexity** is the least number of executions for an algorithm.

Worst case complexity is the maximum number of executions for an algorithm.

(c) 1. B D G H F E C A

2. B , D , G , H

3. 4 or 5

4. A

Topics found Difficult and Confusing

- Reducing a K-map which is filled by ones.
- Coding and decoding of SOP and POS terms.
- Propositional logic and the concept of various operators.
- To obtain the dual and complement of an expression.
- The working to convert POS to SOP form and vice versa.
- Address calculation using Row Major and Column Major.
- Function overriding in inheritance.
- Exceptional handling, Complexity calculations and Big 'O' notation.
- To derive the Maxterms and Minterms from a truth table.
- Drawing logic circuits using Universal gates (NOR , NAND).
- Place/cell values in the K- Maps and marking of Octal , Quads and pairs.
- Recursive techniques, Base case and recursive case.
- Passing objects to the function and returning objects.
- Constructor with inheritance. Use of 'extends ' and 'super' keywords.
- String manipulation programs.
- Algorithms for linked list and other data structures. Deleting a node in a linked list.

Suggestions for Candidates

- Answers and definitions should be short and precise and according to marks allotted.
- Important words and terms should be underlined or highlighted.
- Working should be shown at the side of each question where ever required.
- All the intermediate terms in the truth table must be mentioned.
- Laws must be mentioned while reducing a Boolean Expression.
- Proper labeling should be done in the diagrams / logic circuits.
- Practice one form of K-Map with proper place value for both SOP and POS.
- In programming documentation is compulsory and should be mentioned with each program.
- Declare the class with data members and member functions.
- Expand or define each function according to the instructions given by the side of each function.
- Do not memorize the program, try to understand the logic.
- Practice constructors with every program. Practice programs on the machine.
- Practice algorithms in general or standard form,
- Treat each function of a class as separate program.
- No topic should be left out.