Computer Science (Theory) [Official]
CISCE
ISC
Academic Year: 2023-2024
Date & Time: 18th March 2024, 2:00 pm

Duration: 3h Marks: 70

- 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 you're working and reasoning, wherever required.)

Q1.

1.1. According to the Principle of duality, the Boolean equation

(A+ B') • (A+ 1) =A+ B' will be equivalent to ______.

1. (A' + B) • (A'+ 1) = A' + B
2. (A • B') +(A • 0) = A • B'
3. (A' • B) +(A' • 1) =A' • B
4. (A' • B) +(A' • 0) =A' • B

Solution

According to the Principle of duality, the Boolean equation

A+ B') • (A+ 1) =A+ B' will be equivalent to (A • B') +(A • 0) =A • B'.

Explanation:

The principle of duality asserts that AND becomes OR and vice versa, 0 becomes and vice versa, but complement remains constant.

1.2. When a sequence of OR, NOT, NOR are connected in series, the logic gate obtained is ______.

1. AND
2. NOT
3. OR
4. XOR

Solution

When a sequence of OR, NOT, NOR are connected in series, the logic gate obtained is **OR**.

Explanation:

OR passed through NOT = NOR, NOR passed through NOR= OR.

1.3. Idempotence Law states that ______.

1. X + X = X
2. X + X' = 0
3. X + X = 1
4. X + X' = X

Solution

Idempotence Law states that **X + X = X**.

Explanation:

Idempotence law states that X.X = X and X + X = X.

1.4. Assertion: For proposition ~A=> B, its contrapositive is B =>~A

Reason: Contrapositive is the converse of inverse for any proposition.

1. Both Assertion and Reason are true, and Reason is the correct explanation for the Assertion.
2. Both Assertion and Reason are true, but Reason is not the correct explanation for the Assertion.
3. Assertion is true but Reason is false.
4. Assertion is false but Reason is true.

Solution

Both Assertion and Reason are true, and Reason is the correct explanation for the Assertion.

Explanation:

Inverse of A→B = ~A→~B

Converse is B→A

Contrapositive is ~B→~A

i.e., it is converse of inverse or inverse of converse.

1.5. The complement of the Boolean expression (P' • Q) (R • S') is ______.

1. (P' + Q) • (R' + S)
2. (P + Q') • (R' + S)
3. (P' + Q) • (R + S')
4. (P + Q') • (R + S')

Solution

The complement of the Boolean expression (P' • Q) (R • S') is **(P + Q') • (R' + S)**.

Explanation:

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

1.6.

Assertion: Recursive data structure follows the LIFO principle.

Reason: Execution of recursive code follows the concepts of data structure Queue.

1. Both Assertion and Reason are true, and Reason is the correct explanation for the Assertion.
2. Both Assertion and Reason are true, but Reason is not the correct explanation for the Assertion.
3. Assertion is true but Reason is false.
4. Assertion is false but Reason is true.

Solution

Assertion is true but Reason is false.

Explanation:

Stack follows the LIFO concept, while Queue uses the FIFO principle.

1.7. State any one use of interfaces in Java.

Solution

Interface is used to implement multiple inheritance in Java.

1.8. Write the cardinal form of the maxterm X + Y' + Z.

Solution

Cardinal form of (X + Y' + Z) = $(010)_2$ = 2

1.9. Write the canonical SOP expression for F (A, B) = A <=> B.

Solution

F(A, B) = A . B + A' . B'

1.10. State any one difference between instance variable and class variable.

Differentiate between instance variable and class variable.

Solution 1

1. Each instance variable is created separately, but class variables are created once and shared by all class objects.
2. Instance variables are defined without the static keyword, whereas class variables are declared with the static keyword in a class.

Solution 2

S. No.	Instance variable	Class variable
1.	Instance variables are defined without the 'static' keyword.	Class variables are defined using the Keyword 'static'.
2.	Instance variables are separate copies of an object.	A static variable (class variable) will be accessible as a singular shared instance for all class instances.

Q2.

2.1. Convert the following infix notation to postfix form.

(P + Q * R - S)/T * U

Solution

(P + Q * R - S)/T * U
= (P + QR* - S)/TU*
= (P + QR* S-)/TU*
= (PQR*S-+)/TU*
= PQR*S-+TU*/

I. **Order of precedence as per BODMAS rule:**
()

II. *, /, % Same precedence, but the order of operation is left to right

III. +, - Same precedence, but the order of operation is right to left

2.2. An array ARR [-5 .....15, 10.....20] stores elements in Row Major Wise with each element requiring 2 bytes of storage. Find the address of ARR [10] [15] when the base address is 2500.

Solution

A = B + W * ((I - I_0) * C + (J - J_0))
B = 2500, W= 2, C = 20 - 10 + 1=11, I,J = 10,15 and I_0,J_0 = -5,10
A= 2500 + 2 * ((10-(-5))*11 + (15 - 10))

2500 + 2 * (15*11 +5)
2500 + 2*170
2500 + 340
2840

2.3. The following function is a part of some class:

```
int jolly(int[] x, int n, int m)
 {
if (n<0)
  return m;
else if(n<x.length)
  m=(x[n]>m)?x[n]:m;
return jolly(x, --n, m);
 }
```

a. What will be the output of jolly() when the value of x[ ]={6,3,4,7,1}, n=4 and m=0?

b. What function does jolly() perform, apart from recursion?

Solution

a. 7

Working:

```
jolly({6,3,4,7,1}, 4, 0)
//X.length = 5, n=4, m=0
m=1
jolly({6,3,4,7,1}, 3, 0)
m=7
jolly({6,3,4,7,1}, 2, 0)
m=7
jolly({6,3,4,7,1}, 1, 0)
m=7
jolly({6,3,4,7,1}, 0, 0)
m=7
```

b. Returning the largest no. of the array X[]

2.4. The following function is a part of some class which is used to find the smallest digit present in a number. There are some places in the code marked by ?1?, ?2?, ?3? which must be replaced by an expression/a statement so that the function works correctly.

```
int small_dig(int n)
{   int min=?1?;
    while(n!=0)
    {
    int q=n/10;
    int r=?2?*10;
    min=r>min ? ?3? : r;
    n=q;
    }
    return min;
}
```

a. What is the expression or statement at ?1?

b. What is the expression or statement at ?2?

c. What is the expression or statement at ?3?

Solution

a. n
b. n%10 + 0
c. min

```
int small_dig(int n)
{
int min=n;
while(n!=0)
{
    int q=n/10;
    int r=n%10+0*10;
    min=r>min?min:r;
    n=q;
}
return(min);
}
```

PART II - 50 MARKS

Section-A (Answer any two questions.)

Q3.

3.1. To be recruited as the Principal in a renowned College, a candidate must satisfy any one of the following criteria:

- The candidate must be a Postgraduate and should either possess a B.Ed. degree or a teaching experience of more than 15 years?

OR

- The candidate must be an employee of the same college with a teaching experience of more than 15 years.

OR

- The candidate must be a Postgraduate but not an employee of the same college and should have a teaching experience of more than 15 years.

The inputs are:

INPUTS	
P	Candidate is a Postgraduate
S	Candidate is an employee of the same College
E	Candidate has a teaching experience of more than 15 years
B	Candidate possesses a B.Ed. degree

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

Output: X - Denotes eligibility of a candidate [1 indicates eligibility and 0 indicates ineligibility in all cases]

Draw the truth table for the inputs and outputs given above and write the SOP expression for X (P, S, E, B).

Solution

P	S	E	B	X	Minterm
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0	1	P'.S.E.B'
0	1	1	1	1	P'.S.E.B
1	0	0	0		
1	0	0	1	1	P.S'.E'.B
1	0	1	0	1	P.S'.E.B'
1	0	1	1	1	P.S'.E.B
1	1	0	0		
1	1	0	1	1	P.S.E' .B
1	1	1	0	1	P.S.E.B'
1	1	1	1	1	P.S.E.B

X(P, S, E, B) = P'.S.E.B' + P'.S.E.B + P.S'.E'.B + P.S'.E.B' + P.S'.E.B + P.S.E'.B + P.S.E.B' + P.S.E.B

=∑ (6,7,9,10,11,13,14,15)

3.2. Reduce the above expression X (P, S, E, B) by tiifig 4-variable Karnaugh map, showing the various groups (i.e., octal, quads and pairs).

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

Solution

K-Map

PS\EB	E'.B'	E'.B	E.B.	E.B'
P'.S'				
P'.S			1	1
P.S		1	1	1
P.S'		1	1	1

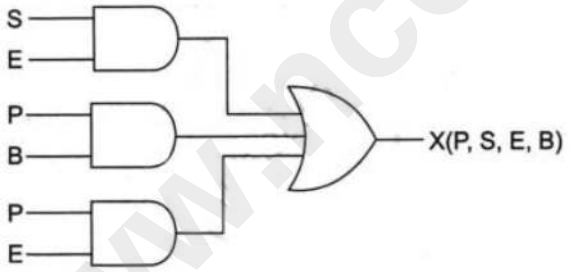
Quad 1(m7,m6,m15,m14) = S.E

Quad 2(m13,m15,m9,m11) = P.B

Quad 3 (m15,m14,m11,m10) = P.E

X(P,S,E,B) = S.E + P.B + P.E

Logic gate



Q4.

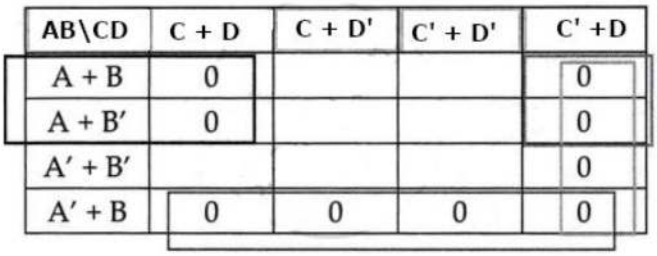
4.1.

4.1. (a) Reduce the Boolean function F (A,B,C,D) = π (0, 2, 4, 6, 8, 9, 10, 11, 14) by using 4-variable Karnaugh map, showing the various groups (i.e., octal, quads and pairs).

Solution

F(A,B,C,D)"= n (0,2,4,6,8,9,10,11,14)

K-Map



AB\CD	C + D	C + D'	C' + D'	C' +D
A + B	0			0
A + B′	0			0
A′ + B′				0
A′ + B	0	0	0	0

Quad 1 (m0,m2,m4,m6) = A+D

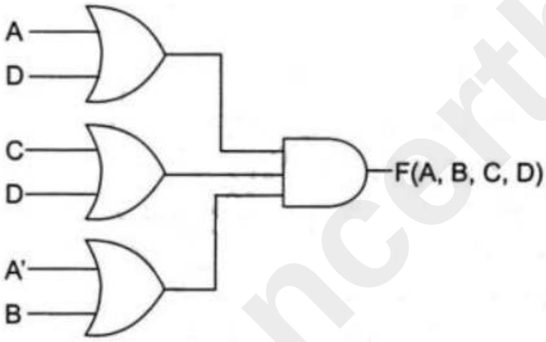
Quad 2 (m2,m4,m14,m10) = C'+D

Quad 3 (m8,m9,m11,m10) =A'+B

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

Solution

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



4.2. Verify if the following proposition is a Tautology, Contradiction or Contingency using a truth table.

((A=>B)^(B=>C))=>(A=>C)

Solution

A	B	C	A→B	B→C	(A→B)^(B→C)	A→C	((A→B^(B→C))→(A→C)
0	0	0	1	1	1	1	1
0	0	1	1	1	1	1	1
0	1	0	1	0	0	1	0

0	1	1	1	1	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	0
1	1	0	1	0	0	0	1
1	1	1	1	1	1	1	1

It is a Contingency.

4.3. Find the complement of the following expression and reduce it by using Boolean laws.

P•(13 ± Q)•Q•(Q+R')

Solution

Complement: (P. (P+Q). Q. (Q+R'))'

P' + (P+Q)' + Q' + (Q+R')' ...[by applying DeMorgan's law]

P' + P' .Q' + Q' + Q' .R

If we further simplify:

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

Q5.

5.1.

5.1. (a) How is a decoder different from a multiplexer?

Solution

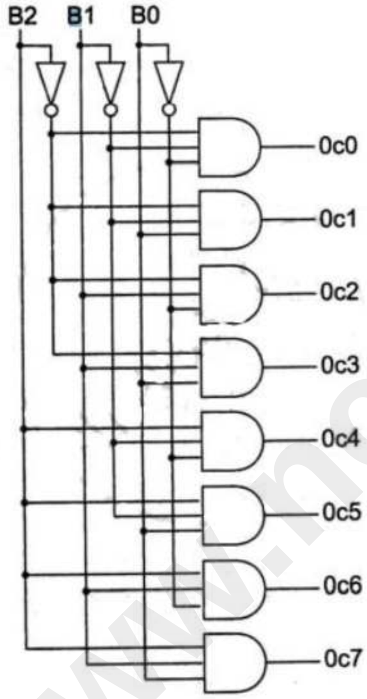
	Multiplexer	Decoder
1.	The multiplexer contains several data input lines and one output line.	The decoder has no data input and several output lines.
2.	There are multiple control lines.	Inputs are the control bits A, B, C, and D.

3.	Depending on the condition of the control line, data from the specific input line is available at the output.	Only one output is high, depending on the state of the control lines.
4.	Converts unary code into binary code.	Converts binary code to unary.

5.1. (b) Draw the logic circuit for 3:8 decoder (Octal decoder). Which multiplexer can be derived from the Octal decoder?

Solution

Decoder circuit

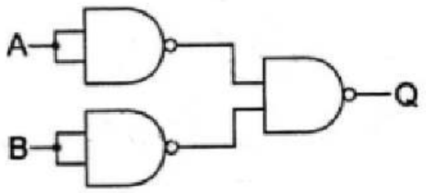


Since, an Octal decoder is 3-to-8 decoder circuit and (2)3 = 8, the said multiplexer will have 8 input lines, 3 select lines and 1 output line. Thus, it will be 8:1 Multiplexer.

5.2. Draw the logic gate diagram for 2-input OR gate using NAND gates only. Show the expression at each Step.

Solution

2-input OR gate using NAND gate:



Working:

Q=A¯.B¯¯

=(A¯)′+(B¯)′ ...(By De Morgan's Law)

= A + B ...(By Involution Rule)

5.3. Write the canonical form of the cardinal terms, m_3 and M_5 for F (A, B, C, D).

Solution

M_3 for F(A,B,C,D) = $(0011)_2$ = A'.B'.C.D (in minterm)

= A+B+C'+D' (in maxterm)

M_5 for F(A,B,C,D) = $(0101)_2$ = A'.B.C'.D (in minterm)

= A+B'+C+D' (in maxtrem)

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)

Q6. Design a class DeciHex to accept a positive integer in decimal number system from the user and display its hexadecimal equivalent.

Example 1:

Decimal number= 25

Hexadecimal equivalent= 19

Example 2:

Decimal number =28

Hexadecimal equivalent = 1C

Some of the members of the class are given below.

Class name	DeciHex
Data members/instance variables:	
num	stores the positive integer
hexa	string to store the hexadecimal equivalent of num
Methods / Member functions:	
DeciHex()	constructor to initialise the data members with legal initial values
void getNum()	to accept a positive integer
void convert(int n)	to find the hexadecimal equivalent of the formal parameter 'n' using the recursive technique
void display()	to display the decimal number and its hexadecimal equivalent by invoking the function convert()

Specify the class DeciHex giving details of the constructor( ), void getNum( ), void convert(int) and void display(). Define a main() function to create an object and call all the functions accordingly to enable the task.

Solution

```
import java.until.Scanner;
class DeciHex
{
 int num;
 String hexa;

 public DeciHex()
 {
  num=0;
  hexa="";
 }
```

```
public void getNum()
{
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter the decimal no. Positive number only");
 num=Math.abs(sc.nextInt());
}

public void convert(int n)
{
 if(n==0)
 return;
 else
 {
  int a=n%16;
  if(a>=0&&a<=9)
  hexa+=""+a;
  else
  {
    switch(a)
    {
    case 10:hexa+='A'; break;
    case 11:hexa+='B'; break;
    case 12:hexa+='C'; break;
    case 13:hexa+='D'; break;
    case 14:hexa+='E'; break;
    case 15:hexa+='F'; break;
   }
  }
  convert(n/16);
 }
}

public void display()
{
 convert(num);
```

```
   System.out.println("Decimal No. "+num);
   System.out.println("Hexa Decimal equialent."+hexa);
  }

  public static void main(String ar[])
  {
   DeciHex ob=new DeciHex();
   ob.getNum();
   ob.display();
  }
}
```

Output:

```
Enter the decimal no. positive number only
123
Decimal No. 123
Hexa decimal equivalent. B7
```

Q7. A class Ins Sort contains an array of integers which sorts the elements in a particular order.

Some of the members of the class are given below.

Class name	**InsSort**
arr[ ]	stores the array elements
size	stores the number of elements in the array
Methods/Member functions:	
InsSort(int s)	constructor to initialise size= s
void getArray( )	accepts the array elements
void insertionSornt( )	sorts the elements of the array in descending order using the in descending order using the Insertion Sort technique

double find( )	calculates and returns the average of all the odd numbers in the array
void display( )	displays the elements of the array in a sorted order along with the average of all the odd numbers in the array by invoking the function find( ) with an appropriate message

Specify the class InsSort giving details of the constructor(), void getArray( ), void insertionSort( ), double find() and void display(). Define a main( ) function to create an object and call all the functions accordingly to enable the task.

Solution

```
import java.util.Scanner;
class InsSort
 {
  int arr[];
  int size;

  public InsSort(int s)
  {
   size=Math.abs(s);
   arr=new int[size];
  }

  public void getArray()
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter"+size+" number for the array");
    for(int i=0;i<size;i++)
     arr[i]=sc.nextInt();
  }
  public void insertionSort()
  {
   for(int i=1;i<size;i++)
   {
    int b=arr[i];
  int j=i-1;
  while(j>=0&& arr[j]<b)
  {
   arr[j+ 1]=arr[j];
   j--;
  }
  arr[j+1]=b;
 }
}
public double find()
{
 int odd=0,c=0
 for(int i=0;i<size;i++)
 {
  if(arr[i]%2!=0)
  {
   odd+=arr[i];
   c++;
  }
 }
 return(odd/c);
}
public void display()
{
  insertionSort();
  System.out.println("Array elements after sorting");
  for(int i=0;i<size;i++)
  System.out.print(arr[i]+" ");
  System.out.println("\nAverage of odd numbers:"+find());
}

public static void main(String args[])
{
  Scanner ins=new Scanner(System.in);
  System.out.println("Enter the size of the array")
  int a=ins.nextInt();
```

```
    InsSort obi=new InsSort(a);
    obi.getArray();
    obi.display();
  }
 }
```

Output:

```
Enter the size of the array
5
Enter 5 number for the array
10 23 12 41 15
Array elements after sorting
41 23 15 12 10
Average of odd number: 26.0
```

Q8. Design a class Coding to perform some string related operations on a word containing alphabets only.

Example: Input: "Java"

Output: Original word: Java

```
J=74
a=97
v= 118
a=97
Lowest ASCII code: 74
Highest ASCII code: 118
```

Some of the members of the class are given below:

Class name	**Coding**
Data members/instance variables:	
wrd	stores the word

len	stores the length of the word
Methods/Member functions:	
Coding()	constructor to initialise the data members with legal initial values
void accept( )	to accept a word
void find()	to display all the characters of 'wrd' along with their ASCII codes. Also display the lowest ASCII code and the highest ASCII code, in 'wrd'
void show()	to display the original word and all the characters of 'wrd' along with their ASCII codes. Also display the lowest ASCII code and the highest ASCII code in 'wrd', by invoking the function find()

Specify the class Coding giving details of the constructor( ), void accept( ), void find( ) and void show(). Define a main() function to create an object and call all the functions accordingly to enable the task.

Solution

```
import Java.util.*;
class Coding
{
 String wrd;
 int len;

 public Coding()
 {
  wrd="";
  len=0;
 }

 public void accept()
 {
  Scanner sc=new Scanner(System.in);
  System.out.println("Enter a word");
  wrd=sc.next(). trim();
 }

 public void find()
 {
  len=wrd.length();
  int small=122,large=65;
  for(int i=0;i<len;i++)
  {
   char ch=wrd.charAt(i);
   if((int)ch>large)
   large=ch;
   else if((int)ch<small)
   small=ch;
   System.out.println(ch+"="+(int)ch);
  }
  System.out.println("Lowest ASCII code:"+small);
  System.out.println("Highest ASCII code:"+large);
 }

 public void show()
 {
  System.out.println("Original word: "+wrd);
  find();
 }

 public static void main(String ar[])
 {
  Coding ob=new Coding();
  ob.accept();
  ob.show();
 }
}
```

Output:

```
Enter a word
```

```
Java
Original word: Java
J = 74
a = 97
v = 118
a = 97
Lowest ASCII code: 97
Highest ASCII code: 118
```

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)

Q9. CardGame is a game of mental skill, built on the simple premise of adding and removing the cards from the top of the card pile.

The details of the class CardGame are given below.

Class name	**CardGame**
Data members/instance variables:	
cards[]	array to store integers as cards
cap	to store the maximum capacity of array
top	to store the index of the topmost element of the array
Methods/Member functions:	

CardGame(int cc)	constructor to initialise cap=cc and top= -1
void addCard(int v)	to add the card at the top index if possible, otherwise display the message “CARD PILE IS FULL”
int drawCard( )	to remove and return the card from the top index of the card pile, if any, else return the value -9999
void display( )	to display all the cards of card pile

i. Specify the class CardGame giving details of the functions void addCard(int) and int drawCard( ). Assume that the other functions have been defined. The main() function and algorithm need NOT be written.

ii. Name the entity described above and state its principle.

Solution

(i)

```
class CardGame
{
 int cards[];
 int cap,top;

 public CardGame(int cc)
 {
  cap= Math.abs(cc);
  top=-1;
  cards=new int[cap];
 }

 void addCard(int v)
 {
  top++;
  if(top==cap)
  {
   System.out.println("CARD PILE IS FULL");
   top--;
  }
  else
  cards[cap]=v;
 }

 public int drawCard()
 {
  if(top==-1)
  {
   System.out.println("CARD PILE IF EMPTY");
   return-9999;
  }
  else
  return cards[cap--];
 }

 public void display()
 {
  if(top==-1)
   System.out.println("EMPTY CARD PILE");
  else
  {
   for(int i=top; i>=0; i--)
   System.out. println(cards[i]);
  }
 }
}
```

(ii) The above entity is a Stack and its principle is Last In First Out (LIFO).

Q10. A super class EmpSal has been defined to store the details of an employee. Defie a subclass (overtime to compute the total salary of the employee, after adding the overtime amount based on the following criteria.

- If hours are more than 40, then 5000 are added to salary as an overtime amount
- If hours are between 30 and 40 (both inclusive), then 3000 are added to salary as an overtime amount
- If hours are less than 30, then the salary remains unchanged The details of the members of both the classes are given below:

Class name	**EmpSal**
Data members/instance variables:	
empnum	to store the name of the employee
empcode	integer to store the employee code
salary	to store the salary of the employee in decimal
Methods/Member functions:	
EmpSal(...)	parameterised constructor to assign values to data members
void show()	to display the details of the employee
Class name	**Overtime**
Data members/instance variables:	
hours	integer to store overtime in hours
totsal	to store the total salary in decimal
Methods/Member functions:	
Overtime(....)	parameterised constructor to assign values to data members of both the classes
void calSal()	calculates the total salary by adding the overtime amount to salary as per the criteria given above

void show()	to display the employee details along with the total salary (salary +overtime amount)

Assume that the super class EmpSal has been defined. Using the concept of inheritance, specify the class Overtime giving the details of the constructer (...), void calSal() and void show().

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

Solution

```
class Overtime extends EmpSal
{
  int hours;
  double totsal;

  public Overtime(String n, int e, double s, int h)
  {
   super(n,e,s);
   hours=h;
  }

  public void calSal( )
  {
   if(hours>40)
     totsal = salary+5000;
   else if(hours>=30)
     totsal = salary+3000;
   else
     totsal = salary;
  }

  public void show( )
  {
    super.show();
    System.out.println("Total salary:"+totsal);
  }
```

}

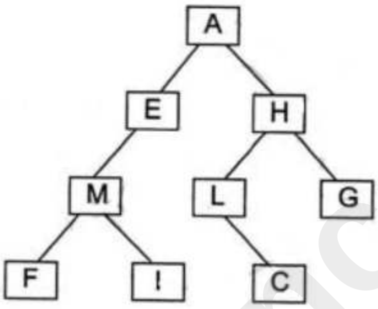
Q11.

11.1. With the help of an example, briefly explain the dominant term in complexity.

Solution

- When determining the performance of an algorithm, we must also consider the term, which is an important component that influences the function's performance. Such a phrase is referred to as dominant.
- For example, let an algorithm have both a nested loop running it N^2 times and a single loop running N times. In this case, N^2 is the dominant term and hence, the complexity is $O(N^2)$.

11.2. Answer the following questions based on the diagram of a Binary Tree given below:



a. Name the external nodes of the tree.

b. State the degree of node M and node L.

c. Write the post-order traversal of the above tree structure.

Solution

a. F, I, C, G

b. Degree of M is 2
Degree of L is 1

c. F, I, M, E, C, L, G, H, A