

SOLUTIONS

SAMPLE QUESTION PAPER - 6

Solved _____

Time : 3 Hours

Maximum Marks : 70

Section A

1. (a)

S. No.	Global Variable	Local Variable
(i)	It is a variable which is declared outside all the functions.	It is a variable which is declared within a compound statement program.
(ii)	It is accessible throughout the program.	It is accessible only within a function/compound statement in which it is declared.

For example,

```
#include<iostream.h>
float Num =9001, // NUM is global void LOCAL (int T)
{
int Total = 0; //Total is a local variable for (int I=0; I<T; I++)
Total +=I;
cout <<Num +Total;
}
void main ()
{
LOCAL (45);
}
```

(1 Mark for Global and Local Variable)

(1 Mark for example)

(b) Keywords are predefined reserved identifiers that have special meanings.

(c) #include <iostream.h>

```
const int Multiple=3;
void main ()
{
int Value =15;
for (int Counter =1; Counter<=5; Counter ++)
{ Value--2
```

```

if (Value % Multiple ==0)
{
cout<<Value*Multiple;
cout<<endl;
};
else
cout<<Value+Multiple<<endl;
}

```

(d) The output of the program is:

Xat@*PVUQVU

[3]

(e) First = 8 Second = 2

[2]

(f) (ii) 1 2 3 4 5 6 7 8 9 10 11

The minimum value of Rndnum is 7.

[2]

2. (a) Copy constructor initializes an instance using the values of another instance of the same type.

Eg.: class student

[½]

```

{
int a;
public:
student () {a = 0;}
student (student &S)
{a = S.a;}
};
void main ()
{
student S1;
Student S2 = S1;    //invokes copy constructor
}

```

(1 mark for definition)

(1 mark for example)

(b) #include<iostream.h>

```

class PAYITNOW
{
int Charge;
public:
void Raise () {cin>>Charge;}
void Show () {cout<<Charge;}
};
void main ()
{
PAYITNOW P;
P. Raise ();
P. Show ();
}

```

(½ mark for each correction)

[2]

```

(c) class COMPETITION
{
    int event-no;
    char description [30];
    int score;
    char qualified;
public:
    void input ()
    {
        cout<<"\n Enter event number";
        cin >>event_no;
        cout<<"\n Enter description";
        gets(description);
        cout<<"\n Enter score";
        cin>>Score;
    }
    void Award (int cutoff score)
    {
        qualified = (score> cut off score? 'Y': 'N');
    }
    void show ()
    {
        cout <<"Event no. : "<<event.no<<"\n Description:"; puts(description);
        cout <<"\n Score:" <<Score<<      "\n Qualified : "<<qualified;
    } };

```

(½ mark for correct syntax for class header)

(½ mark for correct declaration of data member)

(1 mark for correct specification of private member)

(1 mark for appropriate definition of function Award (int cutoffscore))

(1 mark for appropriate definition of function show ()) [4]

(d) (i) DF2 () [1]

(ii) C1, B2, C2, A3, B3 [1]

(iii) 12 bytes [1]

(iv) Data members that are accessible in function F1 () are none because the function F1 () is not the member function of any class. [1]

3. (a) Column major order:

$$= B + W [(I - I_r) + R(I - I_c)]$$

$$B = 6000, W = 8 \text{ bytes}$$

$$M = 20$$

$$\times [16] [16] = 6000 + 8[20(16 - 1) + (16 - 1)]$$

$$= 6000 + 8[20(15) + 15]$$

$$= 6000 + 8[300 + 15]$$

$$= 6000 + 8(315)$$

$$= 6000 + 2520$$

$$= 8520$$

Row Major :

$$\begin{aligned}
 &= B + W [c(I - I_c) + (J - J_c)] \\
 10 \times [10] [2] &= 6000 + 8 [20 (10-1) + (2-1)] \\
 &= 6000 + 8 [20 (9) + (1)] \\
 &= 6000 + 8 [180 + 1] \\
 &= 6000 + 8 [181] \\
 &= 6000 + 1448 \\
 &= 7448
 \end{aligned}$$

Lowest row & column can also be taken as 0. Both answers will be correct.

[1½ + 1½]

(b) void Upper - half (int A [] [], int M, int N)

```

{ for (int i = 0; i < M/2; i++)
  for (int j = 0; j < N/2; j++)
    cout << A [i] [j] << "\n";
}

```

[3]

(c) void delQ(Customer * front)

```

{
  Customer*temp;
  if(front==NULL)
    Cout<<"Queue is empty",
  else
  {
    tem = front;
    front = front->Link;
    delete temp;
  }
}

```

[4]

(d) void addiagonal (int a[8][8], int M, int int N)

```

{
  int Sum=0;
  for(int i=0; i<M;i++)
  for(int j=0; j<N;j++)
  if((i==j) || (i+j==M-1))
  sum=sum + a [i] [j];
  cout << "Sum of diagonal=" << sum;
}

```

[2]

(e)

Symbol	Stack	Expression
A	(A
+	(+	A
((+ (A
B	(+ (AB
*	(+ (*	AB
C	(+ (*	ABC
-	(+ (-	ABC*

Symbol	Stack	Expression
((+ (- (ABC*
D	(+ (- (ABC*D
/	(+ (- (/	ABC*D
E	(+ (- (/	ABC*DE
-	(+ (- (-	ABC*DE/
F	(+ (- (-	ABC*DE/F
)	(+ (-	ABC*DE/F-
*	(+ (-*	ABC*DE/F-
G	(+ (-*	ABC*DE/F-G
)	(+	ABC*DE/F-G*-
*	(+*	ABC*DE/F-G*-
H	(+*	ABC*DE/F-G*-H
)		ABC*DE/F-G*-H*t

Result : ABC*DE/F-G*-H*+

(1 mark for stack status)

[2]

(1 mark for correct expression)

4. (a) Statement 1 to position the file pointer at the beginning of the record is:

```
File.seekg (0);
```

Statement 2 to write the updated Record:

```
File.write ((char*&C, Size of (c));
```

[1]

- (b) void coutspace ()

```
{ fstream fin;
```

```
fin.open ("PARA. TXT", ios ::in);
```

```
char ch [20];
```

```
int count=0;
```

```
if(! fin)
```

```
{
```

```
cout <<"\n can not open file";
```

```
return;
```

```
}
```

```
while (!fin.eof())
```

```
{
```

```
fin>>ch;
```

```
count++;
```

```
}
```

```
cout << "\nNumber of words ="<<count; fin. class ();
```

```
}
```

[2]

- (c) (i) start BookRec

```
{
```

```
char name [20];
```

```
int no.;
```

```
} *Book;
```

```
/* Function to append the record in a file */
```

```

void append()
{
    ofstream bfile;
    bfile.open ("Book.DAT", ios ::app);
    int n, i;
    cout << "Enter how many customers";
    cin>>n;
    for (i=0; i<n;i++)
    {
        cout << "\n Enter the book number:";
        cin >> Book.bno;
        cout<<"\n Enter the book name:";
        gets (Book.name);
        bfile.write ((char*) &Book, size of (Book));
    }
    bfile. close (); }

```

[1½]

(ii) /* Function to search a record in the file */

```

void display ()
{
    ifstream bfile;
    bfile.open ("Book.DAT"; ios :: binary);
    int no, flag;
    flag =0;
    cout <<"Enter book number to be searched:";
    cin >> no;
    while (bfile)
    {
        bfile . read ((char*) & Book. size of (book));
        if (! bfile)
            break;
        if (Book.bno== no)
        {
            cout <<"/n Name"<< Book. Name;
            cout <<"\n Book.Number"<< Book.bno;
            flag=1;
        }
        if(flag==0)
        {cout<<"\n Record not found";
        }
        bfile.close();
    }
}

```

[1½]

Section C

5. (a) **Relation:** (i) Relation is a table, which displays data in a row and column format.

(ii) Each of the relation has a number of columns with unique names.

[1]

Difference between tuple and attribute:

The rows are referred as tuples and columns are referred as attributes.

[1]

- (b) (a) `SELECT * FROM INTERIORS.` [1]
`WHERE TYPE= "Sofa";`
- (b) `SELECT ITEMNAME FROM INTERIORS` [1]
`WHERE PRICE > 10,000;`
- (c) `SELECT ITEMNAME, TYPE` [1]
`FROM INTERIORS WHERE DATEOFSTOCK`
`<{22/01/02} ORDER BY ITEM NAME DESC;`
- (d) `SELECT ITEMNAME, DATE OF STOCK FROM INTERIORS WHERE DISCOUNT>15;` [1]
- (e) (i) 5 [½]
(ii) 13 [½]
(iii) 44,000 [1]

6. (a) **Distributive Law:** This law states that

- (i) $X(Y+Z) = X.Y+X.Z$
(ii) $X+Y.Z = (X+Y). (X+Z)$ [1]

Now, Let us prove using truth table.

- (i) $X.(Y+Z) = X.Y+X.Z$ [½]

X	Y	Z	(Y+Z)	X.(Y+Z)	X.Y	X.Z	(X.Y+X.Z)
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Hence, $X.(Y+Z) = X.Y+X.Z$

- (ii) $X+Y.Z = (X+Y). (X+Z)$ [½]

X	Y	Z	Y.Z	(X+Y.Z)	(X+Y)	(X+Z)	(X+Y). (X+Z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Hence, $X+Y.Z = (X+Y). (X+Z)$

- (b) $(A.C)' + (B.A)' + (B.C)'$ [2]
- (c) The product of sum form of the function $H(V,V,W)$ is:
 $H(V,V,W) = (U+V+W'). (U+V'+W'). (U'+V+W) (U'+V'+W)$ [1]

(d) $F(A,B,C,D) = R(0,1,2,3,8,9,10,11)$.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	1	1	1	1
$\bar{A}B$	0	1	3	2
AB	4	5	7	6
$A\bar{B}$	12	13	15	14
$A\bar{B}$	1	1	1	1
	8	9	11	10

& $F(A,B,C,D) = \bar{A}\bar{B} + A\bar{B}$. [3]

7. (a) Worm is a malicious program like virus. It does not have to attach itself to a host program. [1]
- (b) Two major reasons to have network security are:
- (i) **Secrecy:** Keeping information out of the reach of unauthorized users. [½]
 - (ii) **Authentication:** Determining the authorized user before sharing sensitive information before entering into a business deal. [½]
- (c) **Web Browser:** The Web Browser fetches the page requested, interprets the text and formatting commands that it contains and displays the page property formatted on the screen. [½]
- Example of a Web Browser:** Internet Explorer or Netscape Navigator or Mosaic. [½]
- (d) **Internet:** (i) Internet is a network of networks consisting of thousands of networks throughout the globe.
- (ii) It connects the small networks to prepare a big network. [½]
- Intranet:** Intranet is a network that is the part of Internet. Intranet is a small network which is consisted in Internet. [½]
- (e) (e1) Star topology [1]
- (e2) Wing Z as it has maximum number of computers. [1]
- (e3) (i) **Repeater:** There is no repeater needed as the distance between wings is not very large.
- (ii) **Hub/Switch:** Hub/Switch is needed in every wing to connect the computers. [1]
- (e4) They can use ethernet or telephone connection as it is economic but provides less speed. [1]
- (f) Advantages of E-mail over conventional mailing system:
- (i) E-mail (or Electronic mail) is the electronic way of exchanging information on the internet.
 - (ii) It is the most instantaneous and directly reaches the concerned individual's electronic mail box.
 - (iii) Using e-mail, we can send or receive large volume of information with GUI very fastly and economically. [1]
- (g) **Router:** (i) A Router is a device which is used to connect two or more logical different networks.
- (ii) It accomplishes the connection by organising the large networks into logical network segments. Each of these segments is given a logical address. [1]

●●

SOLUTIONS

SAMPLE QUESTION PAPER - 7

Solved _____

Time : 3 Hours

Maximum Marks : 70

Section A

1. (a) For, INT, New, Name. (½ mark for each identifier)
- (b) cin.getline
abs [1]
- (c) **Corrected code:**
`#include<iostream.h>`
`void main ()`
`{`
`int a, b;`
`cin>>a>>b;`
`int s = sum(a, b);`
`cout<<s;`
`}`
`int sum (int a, int b)` [1]
`{`
`return (a + b);` [1]
`}`
(1 mark for each correction)
- (d) 18, 4 [2]
(The program swaps values of a & b) [1]
- (e) GooDLucK
rreoarD
DLucK [2]
- (f) NORTH : SOUTH : EAST : [1]
SOUTH : EAST : WEST: [1]
2. (a) Data hiding is a mechanism where the details of the class are hidden from the user. The user can perform only a restricted set of operations in a hidden member of the class.
Example:
`class squire`
`{ private:`
`int num;`

```

        public:
        void get ()
        {
        cout<<"Enter Number";
        cin>>Num;
        }
        void display ()
        {
        cout<<"Square is:"<<Num*Num;
        }
        }
        void main ()
        {
        Square Obj;
        Obj.Get ();
        Obj.Display ();
        getch ();
        }

```

[1 mark for difinition]

[1 mark for example]

- (b) (i) Copy constructor

```

        Product P2=P1;
        Product ().check (P2);

```

(ii) Oil:5:5

- (c) class TEST

```

{
private:
int Testcode;
char Description [20];
int NoCandidate;
int CenterReqd;
int ALCNTR(C)
{
return (NoCandidate/100 + 1);
}
void SCHEDULE ()
{
cin>> Testcode;
gets (Description);
cin>>NoCandidate;
cin>>CenterReqd;
NoCandidate = ALCNTR (NoCandidate);
}
void DISPTTEST ()
{
cout <<Testcode;
puts (Description);
}
}

```

```

    cout << NoCandidate;
    cout << CenterReqd;
}

```

[4]

(d) (i) Acode

```

    Aname [20]
    Amount
    Pub [12]
    Turnover.

```

[1]

(ii) BRANCH ()

```

    Haveit ( )
    Giveit ( )

```

[1]

(iii) AUTHOR ()

```

    Start ( )
    Show ( )
    Haveit ( )
    Giveit ( )
    Enter ( )
    Display ( )

```

[1]

(iv) Pub = 12

```

    Turnover = 8
    City = 20
    Employees = 4
    Acode = 2
    Aname = 20
    Amount = 4
    Total = 70 bytes

```

[1]

3. (a) Base Address = 14000 = B.

$$\begin{aligned}
 I[I][J] &= B + W [c(I - Ir) + (J - Jc)] \\
 &= 14000 + 8[10(10 - 0) + (7 - 0)] \\
 &= 14000 + 8 \times 107 \\
 &= 14000 + 856 \\
 &= 14856
 \end{aligned}$$

∴ Location of I[10][7] is 14856

(1 Mark for correct formula).

(1 Mark for putting values in formula correctly).

(1 Mark for correct answer).

[3]

(b) //binary search.(sorted array)

```

int binsearch(int P[20], int VAL)
{
    int Low=0, High=19; int mid;
    int found=0;
    while (low<High&&found!=1)
    {
        mid=(Low + High)/2
        if (P[Mid]==VAL)

```

```

    {
        found=1;
        return found;
    }
    else if VAL<P[mid]
        HIGH = mid-1;
else
    Low=mid+1;
}
    return found;
}

(c) Circular array queue.
void queue::add()
{
    int val;
    cout<<"Enter item to be added";
    cin>>val;
    if(front==0 &&rear==MAX - 1){
        (rear + 1 == front)
        cout<<"Cirucular queue is full";
    }
    else
    {
        if(rear==Max-1)
            rear=0;
        else
            rear++;
        data[rear]=val;
    }
    if(front==-1)
        front=0;
}

void queue::remove C)
{
    if(front==-1)
        cout<<"cirucular queue is empty";
    else
    {
        if (font==rear)
            front=rear=-1;
        else
        {
            if(front==MAX-1)
                front=0;
            else
                front++;
        }
    }
}
}

```

(d) `void colsum (float A [10][10], int r, int C)`

```
{
    int i, j;
    float CS[10];
    for (j = 0; j <= C; j++)
    {CS[j] = 0;
    for (i = 0; i <= r; i++)
    CS[i] += A[i][j];
    }
    for (j = 0; j <= C; j++)
    cout <<CS[j] << "\t";
}
```

[2]

(e)

Symbol	Stack	Output
TRUE		TRUE
FALSE		TRUE, FALSE
TRUE		TRUE, FALSE, TRUE
FALSE		TRUE, FALSE, TRUE, FALSE
NOT	NOT FALSE = TRUE	TRUE, FALSE, TRUE, TRUE
OR	TRUE OR TRUE = TRUE	TRUE, FALSE, TRUE
TRUE		TRUE, FALSE, TRUE, TRUE
OR	TRUE OR TRUE = TRUE	TRUE, FALSE, TRUE
OR	FALSE OR TRUE = TRUE	TRUE, TRUE
AND	TRUE AND TRUE	TRUE

[2]

(1 mark for stack status)

(1 mark for correct answer)

4. (a) `ifile.seekg (10*size of (G));`

[1]

(b) `void count is()`

```
{
    ifstream of("DIALOGUE.TXT");
    char word[20];
    int count=0;
    while(if.eof())
    {
        if>>word;
        if(strem(word,"is")==0)
            count++;
    }
    count<<"No. of is =" <<count;
}
```

[2]

(c) `void createNewfile ()`

```
{
    Game gl;
    ifstream fin;
    ofstream fout;
    fin.open ("GAMES.DAT", ios :: in, ios :: binary);
```

```

fout : open ("BASKET.DAT", ios : : out, ios : : binary);
while (!fin.eof ( ) )
{
fin.read ((char*) & gl, size of (gl));
if (strcmp (gl, Game Name, "Basket Ball") == 0)
{ fout.write ((char * & gl, sizeof (gl));}
}
fin.close ( );
fout.close ( );
}

```

Section C

5. (a) Disadvantages of DBMS: [2]
 (i) High Cost
 (ii) Vulnerable to security threats. [2]
 (b) (a) SELECT Book-Name, Author-Name, Price FROM BOOK WHERE Publishers = "EPB"; [1]
 (b) SELECT Book-Name, FROM BOOK WHERE Type = "Fiction"; [1]
 (c) SELECT Book-Name, Price FROM BOOK ORDER BY PRICE DESC; [1]
 (d) ALTER TABLE BOOK
 SET PRICE = PRICE + 50
 WHERE Publishers = "First Publ."; [1]
 (e) (i) 3 [½]
 (ii) 1,350 [½]
 (iii) My First C++ Brain & Brooke C++ Brain-works A. W. Rossaine
 Fast Cook Lata Kapoor [½]
 (iv) 5 [½]

6. (a) (i)

X	X + 0
0	0
1	1

Verified

(1 mark for correct verification)

(½ mark for any 2 correct columns)

- (ii)

X	X'	X + X'	1
0	1	1	1
1	0	1	1

Hence Verified

(1 mark for correct verification)

(½ mark for any 2 correct columns)

- (b) $UV' + U'V + U'V'$

[2]

- (c)

U	V	W	G	Max Term
0	0	0	1	
0	0	1	1	

0	1	0	0	$U + \bar{V} + W$
0	1	1	0	$U + \bar{V} + \bar{W}$
1	0	0	1	
1	0	1	1	
1	1	0	0	$\bar{U} + \bar{V} + W$
1	1	1	0	$\bar{U} + \bar{V} + \bar{W}$

$$\therefore \text{POS} = (U + \bar{V} + W) \cdot (U + \bar{V} + \bar{W}) \cdot (\bar{U} + \bar{V} + W) \cdot (\bar{U} + \bar{V} + \bar{W}). \quad [1]$$

$$(d) F(U, V, W, Z) = R(0, 3, 4, 5, 7, 11, 13, 15)$$

UV \ WZ	WZ			
	$\bar{W}\bar{Z}$	$\bar{W}Z$	WZ	$W\bar{Z}$
$\bar{U}\bar{V}$	1 0	0 1	1 3	0 2
$\bar{U}V$	1 4	1 5	1 7	0 6
UV	0 12	1 13	1 15	0 14
$U\bar{V}$	0 8	0 9	0 11	0 10

Solving Quad 1 (m_5, m_7, m_{13}, m_{15})

$$= VZ$$

Solving Quad 2 (m_3, m_7, m_{15}, m_{11})

$$= WZ$$

Solving pair (m_0, m_4)

$$= \bar{U}\bar{W}\bar{Z}$$

\therefore Simplified expression is:

$$VZ + WZ + \bar{U}\bar{W}\bar{Z}$$

[1]

7. (a) In linear topology if the bus fails, all the devices get failed.

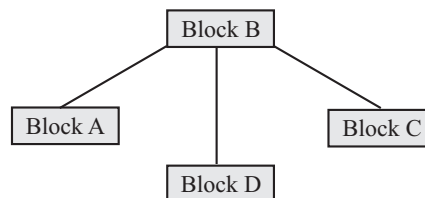
In a star topology if the link between two computers fail, other devices keeps on working. [1]

- (b) Advanced Research Project Agency NETwork is a network system, useful for defence services (in U.S.) It links all the military positions in the country & provides communication path. [1]

- (c) Uniform Resource locator. [1]

- (d) MODEM. [1]

- (e) (i)



(Star Topology)

[1]

- (ii) Block B, because maximum no. of computers are present in it. [1]

- (iii) (a) Repeater: There is no repeater needed as the distance between blocks is not very large. [½]

- (b) Hub/Switch: Hub/Switch is needed in every wing to connect the computers. [½]

- (iv) Use Ethernet Cable. It is cheap and durable. [1]

- (f) 3G is third generation communication technology that deals with faster data speed. [1]

- (g) Switch helps to connect various devices on similar network by managing the traffic. [1]

SOLUTIONS

SAMPLE
QUESTION PAPER - 8

Solved _____

Time : 3 Hours

Maximum Marks : 70

Section A

1. (a) A function prototype tells the compiler the name of the function, the type of data returned by the function, the number of parameters the function expects to receive, the types of the parameters and the order in which these parameters are expected.

example:

```
#include<iostream.h>
int timesTwo (int num); //function prototype
int main ( )
{
    int number, response;
    cout<<"please enter a number:";
    cin>>number;
    response = timesTwo (number); //function call
    cout<<"the answer is"<<response;
    return 0;
} //timesTwo function
int timesTwo (int num)
{
    int answer;
    answer = 2*num;
    return (answer);
}
```

(1 mark for definition)

(1 mark for example)

- (b) #include<iostream.h>

```
#include<iomanip.h>
```

- (c) Correct code:

```
# include<iostream.h>
void main ( )
{
    int P[ ] = {90, 10, 24, 15};
```

[1]


```

int Number = 4, Q = 9;
for (int I = Number-1; I > 0; I --)
switch (I)
{
    case 0 : break;
    case 3 : cout << P [I] << endl;
    case 1 : break;
    case 2 : cout << P[I] + Q;
}
}

```

[2]

(d) 0

21

62

123

[3]

(e) 5,7,5,7

[2]

(f) (ii) RED:BLUE:PINK

[2]

2. (a) Data encapsulation is the property by which several data members and member functions are encapsulated into a single name. [1]

Data Hiding is a property by which several data members are hidden to some member functions. [1]

(b) (i) Constructor

[½]

Factory (Factory P)

[½]

(ii) The Concept illustrated is Constructor Overloading, *i.e.*, polymorphism.

[1]

(c) class RESORT

```

{
    int Rno;
    char Name [10];
    float Charges;
    int Days;
    void Compute ( )
    { if (Days*Charges > 11,000)
      cout <<(Days*Charges);
      else cout<< (1.02*Days*Charges);
    }
    public:
    void Getinfo ( )
    {
        cin >> RNo;
        gets (Name);
        cin >> Charges;
        cin >> Days;
    }
    void Dispinfo ( )
    {
        cout << RNo;
    }
}

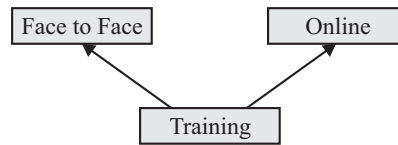
```

```
puts (Name);
cout << Days;
Compute;
}
};
```

[4]

(d) (i) Multiple inheritance

[2]



```
(ii) Input ( );
void Output ( );
siteIn ( );
siteOut ( );
```

[1]

(iii) No, because class Online has no relation to class face-to-face.

[1]

3. (a) $P[R][C] = P[20][30]$ $\therefore R = 20, C = 30$

Element size = 4 bytes

Ir = 0

Jc = 0

 $P[2][20] = 5000, I = 2, J = 20$

To find Base address B

 $P[I][J] = B + W[(I - Ir) + R(J - Jc)]$ $5,000 = B + 4[2 - 0] + 20(20 - 0)$ $= B + 4[2 + 400]$ $= B + 4[402]$ $5,000 = B + 1608$ $B = 5,000 - 1608$ $B = 3492$

[3]

(b) (i) `void avg (int A[], int Size)`

```
{int Sum = 0, avg = 0;
{ for (int i = 0, i < size; i ++ )
{ Sum + = A[i];
avg = Sum/Size;}
} }
```

(ii) `void Hottest (int Month [], int size)`

```
{int Max;
Max = Month [0];
for (int i = 0; i < size; i ++ )
{
if (month [i]>max)
Max=month [i];
}
cout<<Max;
}
```

[1]

```
(iii) void sort (int A[ ], int size)
    { // use selection sort
      for (int i = 0; i < size; i++)
      { if (A[i] > A[i + 1])
        temp = A[i];
        A[i] = A[i + 1];
        A[i + 1] = temp;
      }
      for (int i = 0; i < size; i++)
      {
        cout << A[i];
      }
    }
}
```

[1]

```
(c) #include<iostream.h>
#include<stdio.h>
struct Node
{
    char Country [30];
    Node * Link;
};

class Stack
{ Node * Top;
public:
    Stack ( )
    {
        Top = NULL;
    }

    void Push ( )
    {
        Node * T = new Node;
        gets (T->Country);
        T->Link = Top;
        Top = T;
    }

    void Pop ( )
    {
        if (Top != Null)
        {
            Node * Temp = Top;
            Top = Top->Link;
            delete Temp;
        }
        else
            cout << "Empty Stack";
    }
}
```

```

void Display ( )
{
    Node * Temp = Top;
    while (Temp! = Null)
    { puts (Temp→Country); cout << "\n";
      Temp = Temp→Link;
    }
    ~ Stack ( )
    {
        while (Top != Null)
        {
            Node * Temp = Top;
            Top = Top→Link;
            delete Temp;
        }
    }
}
void main( )
{
    Stack s;
    s. Push ( );
    s. Pop ( );
    s. Disp ( );
}

```

[4]

(1 mark for each function)

(d) int ALTERSUM (int B [] [5], int N, int M)

```

{
    int s = 0, C = 1;
    for (int i = 0; i < N; i ++)
    for (int j = 0; j < M; j ++)
    {
        if (C% 2 != 0)
            s = s + B [i] [j];
        C++;
    }
}

```

[2]

(e)

Element	Stack	Output
TRUE		TRUE
FALSE		TRUE, FALSE
NOT	NOT FALSE = TRUE	TRUE, TRUE
OR	TRUE OR TRUE = TRUE	TRUE
FALSE		TRUE, FALSE
TRUE		TRUE, FALSE
OR	FALSE OR TRUE = TRUE	TRUE, TRUE
AND	TRUE AND TRUE	TRUE

(1 mark for stack status)

(1 mark for correct output)

4. (a) **Statement 1**

```
F. tellr ( );
```

[½]

Statement 2

```
F. Seekp (pos - Sizeof (C));
```

[½]

(b) void TRANSFER ()

```
{ifstream fin;
ofstream fout;
phonlist ph;
fin.open ("PHONE.DAT", ios::in||ios:: binary);
fout.open ("PHONE BACK.DAT", ios::out||ios::binary);
while (!fin.eof())
{ fin.read ((char*) &ph, sizeof (ph));
  if (ph.checkcode ("DEL") == 0)
    fout.write ((char*) & ph, size of (ph));
}
fin.close ();
fout.close ();
}
```

[2]

(c) void Read ()

```
{
FLIGHT F;
ifstream fin;
fin.open ("FLIGHT.DAT", ios : : binary);
while (fin.read ((char*) & F, sizeof (F)))
{
if strcmp (F. Getto ( ), "Mumbai")
== 0)
F. Display ( );
}
fin.close ( );
}
```

Section C

5. (a) **Domain:**

[2]

The domain is a set of possible values that an attribute can have.

[1]

eg. Sex char can have only 2 values

[1]

'M' or 'F'

Domain {'M', 'F'}

(b) (1) (i) SELECT ActivityName FROM

```
ACTIVITY ORDER BY Acode DESC;
```

[1]

(ii) SELECT SUM (Prize Money)

```
FROM ACTIVITY GROUP BY Stadium;
```

[1]

(iii) SELECT Name, Acode FROM COACH ORDER BY Acode;

[1]

(iv) SELECT * FROM ACTIVITY

```
WHERE (Schedule Date < (01-Jan-2004))
```

```
ORDER BY Participants Num;
```

[1]

- (2) (i) 3 [½]
 (ii) 23-June-2004 12-Dec-2003 [½]
 (iii) Ravinder Discuss Throw [½]
 (iv) 1,001 [½]
 1,008
 1,003

6. (a) (i)

X	X'	X . X'	1
0	1	0	0
1	0	0	0

Verified

(1 mark for correct answer) or

(½ mark for any 2 correct columns)

(ii)

X	1	X + 1
0	1	1
1	1	1

Verified

(1 mark for correct answer) or

(½ mark for any 2 correct columns)

(b) $(\bar{U} + V) \cdot (\bar{V} + W)$

[2]

(c)

A	B	C	H	POS
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	1	
0	1	1	1	
1	0	0	1	
1	0	1	0	$(\bar{A} + B + \bar{C})$
1	1	0	0	$(\bar{A} + \bar{B} + C)$
1	1	1	1	

$\therefore \text{POS} = (A + B + C) \cdot (\bar{A} + B + \bar{C}) \cdot (\bar{A} + \bar{B} + C) \cdot (\bar{A} + B + C) \cdot (\bar{A} + B + \bar{C})$

[2]

(d) $F(A, B, C, D) = \Pi(0, 1, 2, 4, 5, 6, 8, 10)$

A B \ C D	C D			
	C + D	C + \bar{D}	\bar{C} + \bar{D}	\bar{C} + D
A + B	0 0	0 1	1 3	0 2
A + \bar{B}	0 4	0 5	1 7	0 6
A + B	1 12	1 13	1 15	1 14
\bar{A} + B	0 8	1 9	1 11	0 10

Quad 1

$$(M_0, M_1, M_4, M_5)$$

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

$$= (A + C)$$

[½]

Quad 2

$$(M_0, M_2, M_4, M_6)$$

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

$$= (A + C)$$

[½]

Quad 3

$$(M_0, M_2, M_8, M_{10})$$

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

$$= (B + D)$$

[½]

$$\therefore \text{Ans} = (A + C).(A + D).(B + D)$$

[½]

7. (a) **Bandwidth:** Bandwidth is defined as the difference between the highest & lowest frequencies that a channel supports.

Unit-Hertz (Hz)

[½ + ½]

- (b) (ii) abps

[1]

- (c) Trojan Horse is a special program that is hidden under same program and works while host program is running.

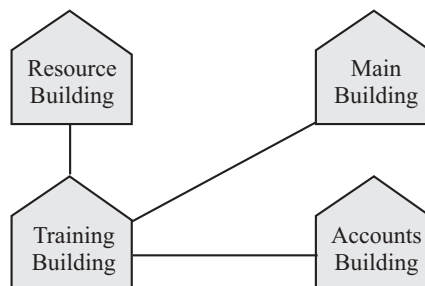
Virus is a stand-alone program.

[1]

- (d) PHP (Hypertext Preprocessor)

[1]

- (e) (e₁) star Topology:



- (e₂) The most suitable place to house the server is training building, because it has maximum computers. [1]

- (e₃) (i) **Repeater:** Repeater is needed between:

1. Training building and Resource building, and
2. Training building and Accounts building, as the distance between them is large.

- (ii) **Hub/switch:** Hub/switch is needed in every building to connect large number of computers. [½ + ½]

- (e₄) Optical Fibre

- (f) (i) Internet Explorer.

[½]

- (ii) Mozilla Firefox.

[½]

- (g) (i) Open Source Software.

[½]

- (ii) Hyper Text Transfer Protocol.

[½]

●●

SOLUTIONS

SAMPLE QUESTION PAPER - 9

Solved _____

Time : 3 Hours

Maximum Marks : 70

Section A

1. (a) typedef is used to assign alternative names to existing types. [1]
e.g. typedef int x;
- (b) `iostream.h` [1]
`ctype.h`
- (c) `#include<iostream.h>` [1]
`void main ()`
`{`
`int X [] = {60, 50, 30, 40}, y;`
`int count = 4;`
`cin >> y;`
`for (int I = count - 1; I >= 0; I--)`
`{`
`switch (I)`
`{`
`case 0 : break;`
`case 2 : cout <<y * x [I] << endl;`
`break;`
`case 1 : break;`
`case 3 : cout <<y +x [I];`
`break;`
`}`
`}` [2]
`}`
- (d) 0*
11*
12*
9*
0*
11*

- 0*
11*
12* [1]
- (e) C # 0
J # 1
H # 1
(1 mark for each correct line of output) [1]
2. (a) Object Oriented Programming approaches the program development in a bottom up fashion while procedural programming approaches in Top Down Fashion.
Object Oriented Programming handles class and objects while procedural handles no objects. [2]
- (b) (i) Exam e1: exam (85); [1]
Exam e2: exam ("XYZ", 96); [1]
(ii) Constructor overloading, *i.e.*, Polymorphism. [2]
- (c)

```
class MOVIE
{
private:
int HALLNO;
char MOVIE-NAME [7]
float WEEK-COLLECTION;
float TOTAL-COLLECTION;
public:
void Read-Data ( )
{
    cin >> HALLNO;
    gets (MOVIE-NAME);
    cin >> WEEK-COLLECTION;
    cin >> TOTAL-COLLECTION;
}
void Display ( )
{
    cout << HALLNO;
    puts (MOVIE-NAME);
    cout << WEEK-COLLECTION;
    cout << TOTAL-COLLECTION;
}
void update ( )
{
    WEEK=WEEK+WEEK-COLLECTION;
    WEEK-COLLECTION=0;
}
};
```

 [2]
- (1 mark for declaration)
(1 mark for each function)

- (d) (i) Multiple inheritance
- (ii) Wall Area, Colorcode, Type, Advance
- (iii) Billing(), Print(), PBook(), Pview(), Book(), View()
- (iv) Exterior(), Paint(), Bill()

3. (a) As Row or column wise storage is not mentioned in the question, assume array A is stored along row wise:

$A[R][C] = A[20][10]$

$R = 20$

$C = 10$

Elements Size = 1 byte

$I_r = 0$

$I_c = 0$

$A[10][5] = 2000$

$I = 10, J = 5$

To find base address of A

$A[I][J] = B + W [C(I - I_r) + (J - I_c)]$

$2000 = CO + 1[10(10 - 0) + (5 - 0)]$

$2000 = CO + [100 + 5]$

$2000 - 105 = CO$

$1895 = CO$

\therefore base Address = 1895.

[3]

- (b) `void Lower_half (int a [] [10], int N)`

```
{
    int i; j;
    for (i = 1/2, i < N; i++)
    {
        if (j = 0, j < N; j++)
        {
            if (i >= j)
                cout << a [i] [j] << " ";
            else
                cout << " ";
        }
        cout << "\n";
    }
}
```

[3]

- (c) `void DELETE (NODE*front)`

```
{
    NODE * ptr = front;
    if (ptr == NULL)
    {
        cout << "UNDERFLOW",
        exit ( );
    }
    else if (front == rear)
```

```

{
    delete ptr;
    front = rear = NULL;
}
else
{ front = front->Link;
  delete ptr;
}
}

```

[4]

(d) void Diagonal (float A [10] [10], int R, int C)

```

{
    int i, j;
    for (i = 0; i < R; i ++)
    {
        for (j = 0; j < C; j ++)
        { if (i == j)
          { cout << A [i][j];
            cout << "/n";
          } }
    } }
} }

```

[2]

(e)

Step	Input Symbol/Element	Stack	Intermediate Calculations Output
1.	Push 5	5	
2.	Push 6	5, 6	
3.	Push 2	5, 6, 2	
4.	+ Pop (2 elements) & evaluate	5	$6 + 2 = 8$
5.	Push result (8)	5, 8	
6.	* Pop (2 elements) & evaluate	# empty	$5 * 8 = 40$
7.	Push result 40	40	
8.	Push 12	40, 12	
9.	Push 4	40, 12, 4	
10.	/ Pop (2 elements) & evaluate	40	$12/4 = 3$
11.	Push result (3)	40, 3	
12.	- Pop (2 elements) & evaluate	#	$40 - 3 = 37$
13.	Push result (37)	37	
14.	No-more-elements		37 (result)

(1 mark for stack status, 1 mark for final result)

[5]

4. (a) File.seekp ((REC)* sizeof (L)); //statement 1.
 File.write ((char*) & L, size of (L)); //statement 2.

[1]

(b) void countline ()
 { if streamfil ("STORY.TXT");
 int LINES = 0;
 char STR[80];
 while (fil.getline (STR, 80))
 LINES ++;

```

        cout <<"No. of lines:" << LINES <<"\n";
        fil.close ( );
    }
(c) void COPYABC ( )
{
    ifstream fin;
    ofstream fout;
    Directory ph;
    fin.open ("TELEPHONE.DAT", ios : in ios : : binary);
    fout.open ("TELEPHONE.DAT", ios : : out ios : : binary);
    while (! fin.eof ( ) )
    {
        fin.read ((char*) & ph), Sizeof (ph));
        if (ph.checkcode ("123") == 0)
            fout.write ((char*) & ph, sizeof (ph));
    }
    fin.close ( );
    fout.close ( );
}

```

[3]

Section C

5. (a) Hierarchical Data Model :
- (i) Data is represented as a tree.
 - (ii) Each data has parent child relationship.
 - (iii) A child can have only one parent. [2]
- (b) (a) SELECT * FROM FURNITURE WHERE TYPE = "BABYCOT"; [1]
- (b) SELECT ITEMNAME FROM FURNITURE WHERE PRICE > 15,000; [1]
- (c) SELECT ITEMNAME, TYPE FROM FURNITURE WHERE DATE_OF_STOCK < {22/01/11}
ORDER BY ITEMNAME DESC; [1]
- (d) SELECT ITEMNAME, DATE_OF_STOCK FROM FURNITURE WHERE DISCOUNT > 25; [1]
- (e) (i) 5 [½]
- (ii) 25 [½]
- (iii) 17.5 [½]
- (iv) 57,000 [½]
6. (a) $X + X'Y = X + Y$
 $\text{LHS}(X + X')(X + Y)$ [Distributive law]
 $\& 1. (X + Y)$ [X + X' = 1]
 $\& (X + Y) = \text{RHS}$ [2]
- (b) $(A \cdot C)' + (A' \cdot B')' + (B' \cdot C)'$
 $= A' + C' + A + B + B + C$
 $= 1 + 1 + B$
 $= 1.$
- (c) $F = \begin{matrix} AB'C & ABC' & ABC \\ 101 & 110 & 111 \end{matrix}$
 $= \Sigma(5, 6, 7)$
 $= \pi(0, 1, 2, 3, 4)$

$$\begin{aligned}
 &= (A + B + C) \cdot (A + B + C') \cdot (A + B' + C) \\
 &= (A + B' + C') (A' + B + C) \\
 \text{(d) } F &= R(4, 5, 7, 9, 11, 12, 13, 14, 15)
 \end{aligned}$$

		c d			
		$\bar{c} \bar{d}$	$\bar{c} d$	$c d$	$c \bar{d}$
a b	$\bar{a} \bar{b}$	0	0	0	0
	$\bar{a} b$	1	1	1	0
a b	$a \bar{b}$	1	1	1	1
	$a b$	0	1	1	0

[1]

Solving Quad 1: $m_4 + m_5 + m_{12} + m_{13}$

on solving we get $\bar{c}b$

[½]

Solving Quad 2: $m_5 + m_7 + m_{13} + m_{15} = bd$

Solving Quad 3: $m_{12} + m_{13} + m_{14} + m_{15} = ab$

[½]

Solving Quad 4: $m_9 + m_{11} + m_{14} + m_{15} = ad$

[½]

\therefore The reduced expression is:

$$b\bar{c} + ab + ad + bd$$

7. (a) In BUS Topology if bus fails, then all the devices fails.

In STAR, if one connection fails, it does not disturb other devices.

[1]

(b)

HTML	XML
(i) Presentation language.	Data Storage language.
(ii) No Need of Parsing.	Parsing is Required.

[2]

- (c) Cyber Law regulates illegal use of internet.

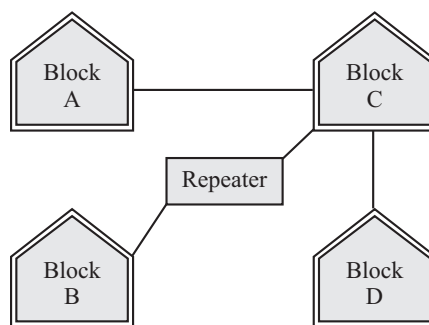
[1]

- (d) Local Area Network

OSI = Open System Interconnection.

[1]

- (e) (e₁) Star Topology



- (e₂) Block C, because maximum no. of computers are present in it.

[1]

- (e₃) (i) **Repeater:** Repeater should be placed between block C and block B as there is large distance between them.

[½]

- (ii) **Hub/switch:** Hub/switch is needed in every block to connect the computers.

[½]

- (e₄) Dial-up or Telephone link

[1]

- (f) Uninterrupted Communication.

[1]

- (g) Firewall (router).

[1]

●●

SOLUTIONS

SAMPLE QUESTION PAPER - 10

Solved _____

Time : 3 Hours

Maximum Marks : 70

Section A

1. (a) #include is used to:
 - (i) include built in header files. [1]
 - (ii) include external files [1]
- (b) #include<iostream.h>
 void main ()
 {
 int X;
 float F;
 cin >> x;
 if(x < 2)
 F = 2 * x * x + 5;
 if (x == 0)
 F = 0;
 else
 F = - (2 * x * x + 5);
 cout << F;
 } [2]
- (c) (i) 2 times [1]
 (ii) 14.
- (d) 10*
 2*
 10#
 2#
 12#
 6#
 10@
 2@
 12@ [3]

- (e) Col 40 Row 50
Col 50 Row 40
Col 60 Row 70 [2]
- (f) Max Value = 19
Min Value = 10 [2]
2. (a) Visibility modes are the methods by which access to certain members is controlled from others. [1]
Eg. public, private, protected. [1]
- (b) @@@@
9 18 27
Polymorphism (or) Function overloading is the feature used in the following program. [2]
- (c)

```
class HOUSING
{
    int REG-NO;
    char NAME [20];
    char TYPE;
    float COST;
public:
    void Read-Data ( )
{ cin >> REG-NO;
  gets (NAME);
  gets (TYPE);
  cin >> COST;
}
    void Display ( )
{
    cout << REG-NO;
    puts (NAME);
    putc (TYPE);
    cout << COST;
}
    int Draw-NOS ( )
{
    randomize ( );
    int a [10];
    a = random (2);
    if (a == REG-NO)
        return 1;
    else
        return 0;
}
};
```

 [4]

- (d) (i) No member is directly accessible. [1]
 (ii) 24 bytes. [1]
 (iii) `Read-sofa-details (). Disp-office-details ().`
`Disp-sofa-details ().`
`Read-fur-details ().`
`Disp-fur-details ().` [2]

3. (a) (i) When Stored in column Major:

`S [R] [C] = S [15] [10]`
`R = 15`
`C = 10`
 Element Size (W) = 4
 Base address (B) = 2,000
`S [8] [9] = S [I] [J]`
`I = 8, Ir = 0`
`J = 9, Jc = 0`
`S [8] [9] = B + W ((I – Ir) + R(J – Jc))`
`S [8] [9] = 2000 + 4((8) + 15(9 – 0))`
`S [8] [9] = 2000 + 4(8 + 135)`
`= 2000 + 4 × 143`
`= 2000 + 572`
`= 2572`

[1½]

(ii) When Stored in Row Major:

`S [I] [J] = B + W (C(I – Ir) + (J – Jr))`
`= 2000 + 4 (10 (8) + (9))`
`= 2000 + 4 (180 + 9)`
`= 2000 + 4 (189)`
`= 2000 + 4 × 189`
`= 2000 + 756`
`= 2756`

[1½]

(b) `void prod (int A [5] [5], int r, int c)`

```
{
    int P[r], i, j,
    for (i = 0, i < r, i++)
    {
        P [i] = 1;
        for (j = 0, j < c, j++)
            P [i] * = A [i] [j];
        cout << "Product of Row" << i + 1 <<
        < " = " << P [i] << endl;
    }
}
```

[3]

(c) `#include <iostream.h>`

`#include <stdio.h>`

`struct node`

```
{
    double a;
```



```

    Node*Link;
}
class Stack
{
    Node*Top;
public:
    Stack ( )
    {
        Top = NULL;
    }
    void Pop ( )
    {
        if (Top! = NULL)

        {
            Node*Temp = Top;
            Top = Top->Link;
            delete Temp;
        }
    }
    void main ( )
    {
        Stack s;
        s. Pop ( );
    } }

```

[4]

(d) Algorithm to insert an element in queue

1. Check if the queue is full.
2. Increment $\text{Rear} = \text{Rear} + 1$
3. Insert Element
 $\text{Queue}[\text{Rear}] = \text{Item}$
4. Set Front to $\text{Rear} + 1$

[2]

(e)

Symbol	Stack	Output
20		20
10		20, 10
+	$20 + 10 = 30$	30
5		30, 5
2		30, 5, 2
*	$5 * 2 = 10$	30, 10
-	$30 - 10 = 20$	20
10		20, 10

[2]

4. (a) (i) `File.seekp (-1*size of (s), ios::cur);`
 (ii) `File.write ((char*) &S, size of (s));`

[1]

(b) `void Countnu()`

```

{
    if stream f("CONTENT.TXT");
    char ch;

```

```

    int count=0;
    while (!f. eof())
    {
        f>>ch;
        switch(ch)
        {
            case 'N':case 'n':
            case 'U':case 'u':
                count++;
        }
    }
    f. close();
    cout<<count;
}
(c) void Addnew ()
{ fstream F;
  STUD S;
  F.open ("STUDENT. DAT", ios :: binary);
  S.Enter ();
  F.write ((char*) & S, sizeof (S));
  F.close ();
}

```

[3]

Section C

5. (a) DDL-Data Definition Language. [1]
 DDL-Data Manipulation Language [1]
- (b) (i) SELECT Sender Name FROM SENDER
 WHERE Sender City = "MUMBAI"; [1]
- (ii) SELECT Rec ID, Sender Name, Sender Address, Rec Name, Rec Address
 FROM SENDERS, RECIPIENT R
 WHERE S. SENDER ID = R. SENDERID; [1]
- (iii) SELECT * FROM RECIPIENT
 ORDER BY RecName; [1]
- (iv) COUNT (*) FROM RECIPIENT
 GROUP By Rec City; [1]
- (v) NEW DELHI
 MUMBAI [½]
- (vi) Rjain HSingh [½]
- (vii) R Bajpayee 5, central Avenue
 H Singh 2A, Andher East
 PK Swamy B5, CS Terminus [½]
- (viii) ND08 S MAHAJAN
 ND48 S Tripathi [½]

6. (a) $X + Y.Z = (X + Y).(X + Z)$

X	Y	Z	Y.Z	X+Y.Z	X+Y	X+Z	(X+Y).(X+Z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

(2 marks for correct truth table)

Note : No marks are to be awarded if only algebraic proof is given.

(b) $U'V + UV'$

[2]

(c)

x	y	z	f	minterm
0	0	0	0	
0	0	1	1	$\bar{x}\bar{y}z$
0	1	0	0	
0	1	1	0	
1	0	0	1	$x\bar{y}\bar{z}$
1	0	1	1	$x\bar{y}z$
1	1	0	0	
1	1	1	1	xyz

$\therefore \text{SOP} = \bar{x}\bar{y}z + x\bar{y}\bar{z} + x\bar{y}z + xyz$

[1]

(d) $F(u, v, w, z) = R(0, 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14).$

	$\bar{w}\bar{z}$	$\bar{w}z$	$w\bar{z}$	wz
$\bar{u}\bar{v}$	1 0	1 1	1 3	1 2
$\bar{u}v$	1 4	1 5	1 7	1 6
uv	1 12	1 13	1 15	1 14
$u\bar{v}$	1 8	1 9	1 11	1 10

$\Rightarrow f(u, v, w, z) = \bar{u}\bar{v} + \bar{u}\bar{w} + w\bar{z}u + u\bar{v} + \bar{u}z$

[3]

7. (a) Mbps.

[½]

Mega bits per second.

[½]

(b) Cookies are files that store user login data on a website.

[½]

Firewall is a software that controls the unauthorised network access to a computer.

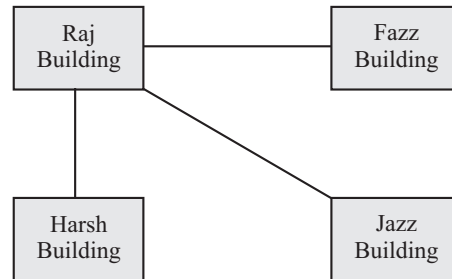
[½]

(c) Chatting is a method of communicating with other people by direct messages, voice or videos by the use of internet.

(d) Ethernet is a cabling technique standard to connect network devices.

[1]

(e) (e₁) Star topology:



[1]

(e₂) The server should be placed in Raj Building as it has maximum number of computers in it. [1]

(e₃) (i) **Internet connecting device/Modem:** It is required in every building for converting digital signal to analog & analog signal to digital to be displayed on the screen. [½]

(ii) **Switch:** It is required in every building to connect various computers for internet. [½]

(e₄) MAN, as it is economical than WAN & converts large area than LAN. [1]

(f) Web Hosting is a method of online storing of files. [1]

(g) Data channel is a path through which data is transferred over network. [1]

●●