1. (a) Local Variables: Local variables are those variables which are declared within a function or a compound statement and these variables can only be used within that function/scope.

Global Variables: Global variables are those variables which are not declared within any function or scope. So, these variables can be accessed by any function of the program.

Example

```c++
#include <iostream.h>
#include <conio.h>

int G; // Global variable declared

void F() {
    int L = 25;  // Local variable of function F() assigned value 25
    G = 5;      // Global Variable is accessed and assigned value 5
    cout << G << endl; // Value of global variable is displayed as 5
    cout << L << endl; // Value of local variable is displayed
}

void main() {
    F();       // Function call
    G = G + 5; // Global variable is incremented by 5
    cout << G << endl;  // Global variable is displayed as 10
}
```

(½ Mark for each correct explanation of Local Variable and Global Variable)

(½ Mark for each correct example of Local variable and Global Variable) OR

(Full 2 Marks for correct example(s) demonstrating the meaning of / difference between Local Variable and Global Variable)

OR

(Only 1 Mark to be awarded if Explanation without supporting examples)
(b)

```cpp
#include<iostream.h>
#include<string.h>
typedef char Text [80];
void main()
{
    Text T = "Germany";
    int Count = strlen(T);
    cout << T << " has " << Count << " characters" << endl;
}
```

(d) 30
21#42
30#240
250#

(½ Mark for each correct value)

Note:
Deduct ½ Mark for not writing # at proper places
Deduct ½ Mark for not considering endl at proper places

(e)

S 10
Next@50
Last@40
Reset to 0

(½ for writing each line of output)
2. (a)

Private members of a class are accessible only to the member functions of the same class.

Public members of a class are accessible to the member functions of the same class as well as member functions of its derived class(es) and also to an object of the class.

Example:

```cpp
class Base
{
    int N;
public:
    void Assign()
    {
        N=10;
    }
};

class Derived: public Base
{
    int X;
public:
    void DisplayBase()
    {
        cout<<N; //Not Accessible
        Assign(); //Accessible
    }
};
```


```cpp
void main ( )
{
    Base B;
    B.Assign( ); //Accessible
}
```

(1 Mark for correct explanation OR example illustrating non accessibility of Private Members inside Derived class)

(1 Mark for correct explanation OR example illustrating accessibility of Public Members inside Derived Class and to object of the class)

(b)

@ @ @ @
91827
Polymorphism
OR
Function Overloading

(½ Mark for writing each correct line of output)

(1 Mark for writing the feature name correctly)

(c)

class Candidate
{
    long RNo;
    char Name[20];
    float Score;
    char Remarks[20];
    void AssignRem();

public:
    void Enter( );
    void Display( );
} ;
void Candidate: :AssignRem( )
```
if (Score>=50)
    strcpy (Remarks,"Selected") ;
else
    strcpy(Remarks,"Not Selected") ;
}

void Candidate::Enter ( ) {
    cin>>RNo ;
    gets (Name) ; cin>>Score;
    AssignRem( ) ;
}

void Candidate: :Display()
{
    cout<<RNo<<Name<<Score<<Remarks<<endl;
}

(½ Mark for correct syntax for class header)
(½ Mark for correct declaration of data members)
(1 Mark for correct definition of AssignRem())
(1 Mark for correct definition of Enter() with proper invocation of AssignRem() function)
(1 Mark for correct definition of Display())

NOTE:

Deduct ½ Mark to be deducted if Assignrem() is not invoked properly inside Enter( ) function

No marks to be deducted if member function definitions are written inside the class

(d) (i) Multiple Inheritance
(1 Mark for correct answer)

(ii) CCode, CourseName, StartDate, EndDate, Pay
(1 Mark for correct answer)

Note: No marks to be awarded for any other alternative answer
(iii) Commence( ), CDetail( ), Register( ), Display( )

(I Mark for correct answer)

Note: No marks to be awarded for any other alternative answer

Constructor functions to be ignored

(iv) Enter( ), Show( )

(I Mark for correct answer)

Note: No marks to be awarded for any other alternative answer

Constructor functions to be ignored

3(a) void func(int A[], int B[], int C[], int N, int M, int &K)
{
    int I=0, J=M-1;
    K=0;
    while (I<N && J>=0)
    {
        if (A[I]<B[J])
            C[K++]=A[I++];
        else
            if (A[I]>B[J])
                C[K++]=B[J--];
            else
                {
                    C[K++]=A[I++];
                    J--;
                }
    }
    for (int T=I; T<N; T++)
        C[K++]=A[T];
    for (T=J; T>=0; T--)
        C[K++]=B[T];
}

(½ Mark for function header with desired parameters)
(½ Mark initializing counters)
(1 Mark for correct formation of loop)
(1 Mark for correct comparison of elements)

(I Mark for transferring remaining elements in resultant array)

(b) Assuming  LBR=LBC=0

B=5200

W=4 bytes

Number of Rows(N)=20

Number of Columns(M)=50

LOC(Arr[I][J]) = B + (I + J*N)*W
LOC(Arr[15][10]) = 5200 + (15 + 10*20)*4

= 5200 + (215*4)
= 5200 + 860
= 6060

(1 Mark for writing correct formula (for row major) OR substituting formula with correct values for calculating Address)

(2 marks for calculating correct address)

Note:
1 Mark to be awarded for writing only the correct answer (i.e. 6060)

(c)

void Queue: : QueInsert()
{
    NODE* Temp = new NODE;
    cin >> Temp->Pno; gets (Temp->Pname);
    Temp->Link = NULL;
    Rear->Link = Temp;
    Rear = Temp;
}

(1 Mark for creating a new NODE dynamically)

(1 Mark for assigning NULL to Link of new NODE)

(1 Mark for linking the Rearmost NODE to the new NODE)

(1 Mark for making the new NODE as the Rearmost NODE)

(d)

void COLSUM(int A[] [100], int N, int M)
{
    int SUMC;
    for (int j=0; j<M; j++)
    {
        SUMC = 0;
        for (int i=0; i<N; i++)
SUMC = SUMC + A[i][j];
Cout<<"Sum of Column "<<j<<" = "<<SUMC;
}
}

(½ Mark for writing correct outer loop)
(½ Mark for initializing SUMC with 0 for each column)
(½ Mark for writing correct inner loop)
(½ Mark for finding sum of each column)

(e) 1100

(2 Mark for evaluating 1100 as the final answer)

Note:
(½ Mark for writing only the final answer as 1100 without showing the operations or Stack)

4(a)    File.seekg(0,ios::end);  //Statement 1
        File.tellg();            //Statement 2

( ½ Mark for each correct statement)

(b) void COUNT ( )
{
    ifstream Fil ("Mine. TXT");
    char STR [10] ;
    int count = 0 ;
    while (!Fil.eof ( ))
    {
        Fil>>STR;
        if (strcmp (STR, "Me") ==0 || strcmp(STR,My")==0)
            count++;
    }
    Cout<<"Count of Me/My in file : "<<count<<end1;
    Fil.close( ) ; //Ignore
Any other correct function definition performing the desired operation

(½ Mark for opening DIARY.TXT correctly)

(½ Mark for reading each word (Whichever method adopted) from, the file)

(½ Mark for comparing the word with 'Me' and 'My' and incrementing counter)

(½ Mark for displaying the number of 'Me/My' with/without the Text Message)

NOTE:
Ignore case sensitivity check for Me/My

(c) void Search()

{LAPTOP L;

long modelnum;

cin >> modelnm;

ifstream fin;

fin.open ("LAPTOP.DAT", ios: :binary | ios: :in);

while (fin.read ((char*) &L, sizeof (L)))

{if (L.ReturnModelNo ( ) == modelnum)

L.StockDisplay ( ) ;

} Fin.close () ; //Ignore

}

OR

void Search (long modelnum)

{

LAPTOPL;

ifstream fin;

fin.open ("LAPTOP.DAT", ios: :binary | ios: :in) ;

while (fin.read ((char*) &L, sizeof (L)))
if (L.ReturnModelNo() == modelnum)
    L.StockDisplay();
}
fin.close(); //Ignore

5(a) Cartesian Product (binary operator): It operates on two relations and is denoted by X. For example Cartesian product of two relations R1 and R2 is represented by R = R1 X R2. The degree of R is equal to sum of degrees of R1 and R2. The cardinality of R is product of cardinality of R1 and cardinality of R2.

Union (binary operator): It operates on two relations and is indicated by U. For example, R=R1 U R2 represents union operation between two relations R1 and R2. The degree of R is equal to degree of R1. The cardinality of R is sum of cardinality of R1 and cardinality of R2.

Following have to be considered for the operation R1 U R2.
Degree of R1 = Degree of R2
jth attribute of R1 and jth attribute of R2 must have a common domain.

(b) (i) SELECT * FROM WORKER ORDER BY DOB DESC;

(1 Mark for correct query)
(½ Mark for partially correct answer)
(ii) SELECT NAME, DESIG FROM WORKER  
WHERE PLEVEL IN ('P001', 'P002');  
OR  
SELECT NAME, DESIG FROM WORKER  
WHERE PLEVEL = 'P001' OR PLEVEL= 'P002';  
(1 Mark for correct query)  
(½ Mark for partially correct answer)

(iii) SELECT * FROM WORKER  
WHERE DOB BETWEEN '19-JAN-1984' AND '18-JAN-1987';  
OR  
SELECT * FROM WORKER  
WHERE DOB >= '19-JAN-1984' AND DOB<='18-JAN-1987';  
OR  
SELECT * FROM WORKER WHERE DOB > '19-JAN-1984' AND DOB<'18-JAN-1987' ;  
(1 Mark for correct query)  
(½ Mark for partially correct answer)

(iv) INSERT INTO WORKER  
VALUES (19, 'Daya Kishore', 'Operator' , 'P003' , '11-Jun-1984');  
(1 Mark for correct query)  
(½ Mark for partially correct answer)

(c) (i)   COUNT (PLEVEL)   PLEVEL  
1        P001  
2        P002  
2        P003  
( ½ Mark for correct output)  
(ii)
12-Jul-1987 13-Sep-2004

({½ Mark for correct output})

(iii) Name    Pay
       Radhe Shyam  26000
       Chander Nath 12000

({½ Mark for correct output})

(iv)

P003  18000

({½ Mark for correct output})

6(a) \[ X + XY = X \]

\[
\begin{align*}
\text{L.H.S} & = X + XY \\
& = X(1 + Y) \\
& = X \\
& = \text{R.H.S}
\end{align*}
\]

Since the second law is dual of first and first has been proved, by duality principle second also holds true.

(1 mark for stating the correct law)
(1 mark for the appropriate verification using algebraic method)

(b) \[ F = (X+Y')(X+Z') \]

(2 Marks for the final expression \((X+Y')(X+Z')\))

OR

(1 Mark for anyone of the correct terms out of \((X+Y')\) or \((X+Z')\))

(c) \[ F(A,B,C) = (A+B+C)(A+B'+C')(A'+B+C')(A'+B'+C) \]

(1 Mark for the correct POS form)

Note: Deduct ½ mark if wrong variable names are used

(d) \[ F(P,Q,R,S) = P'R' + R'S' + P'S' \]

(½ Mark for placing all 1s at correct positions in K-Map)
(½ Mark for each grouping)
(1 Mark for writing final expression in reduced/minimal form)
Note: Deduct ½ mark if wrong variable names are used

7(a) URL stands for Uniform Resource Locator. Each page that is created for Web browsing is assigned a URL that effectively serves as the page’s worldwide name or address. URL’s have three parts: the protocol, the DNS name of the machine on which the page is located and a local name uniquely indicating the specific page (generally the filename).

(1 Mark for correct significance)

(b) Bus Topology

<table>
<thead>
<tr>
<th>Slower as compared to star topologies of network topology</th>
<th>Star Topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expensive as compared to Bus topology</td>
<td></td>
</tr>
<tr>
<td>Breakage of wire at any point disturbs the entire network</td>
<td>Long wire</td>
</tr>
<tr>
<td></td>
<td>Length</td>
</tr>
</tbody>
</table>

(½ Marks for written or diagrammatic explanation of correct Bus Topology)

(½ Marks for written or diagrammatic explanation of correct Star Topology)

(c) Voice over Internet Protocol (Voice over IP, VoIP) is communication protocols and transmission technologies for delivery of voice communications and multimedia sessions over Internet Protocol (IP) networks, such as the Internet. Also, we can say, VoIP are IP telephony, Internet telephony and broadband telephony.

(1 Mark for explaining VoIP)

Note: 1 Mark for writing full form of VoIP

(d) Java script & VB script are client side scripts

JSP & ASP are server side scripts

(½ Mark for writing correct Client Side Scripts)

(½ Mark for writing correct Server Side Scripts)

(e)
e1) HR center:

Reason as it has the maximum number of computers

OR

Business Block

Reason as it is closest to all other Centers (minimum cable length required)

(½ Mark for mentioning any correct place)

(½ Mark for correct reason)

OR

(1 Mark for any other location with a valid reason)

(e2)

Law block---------→Business block------------------→Technology block--------→HR center

(1 Mark for mentioning any valid connectivity or topology or diagram connecting various compounds inside the campus)

(e3) Switch

(1 Mark for mentioning correct device)

Note:

(1 mark to be awarded if Switch / Hub is mentioned)

(e4) WAN as the given distance is more than the range of LAN and MAN.

(1 Mark for correct network type)

(f) (ii) and (iii)

Note:

No marks to be awarded for any other alternative answer

(½ Mark for each correct option)

(g) Open source software refers to a program or software in which the source code (the form of the program when a programmer writes a program in a particular programming language) is available to the general public for use and/or modification from its original design free of charge.

Proprietary software is software that is owned by an individual or a company (usually the one that developed it). There are almost always major restrictions on its use, and its source code is almost always kept secret.

(½ Mark for each correct explanation/comparison)