SPECIAL STUDY MODULE

COMPUTER SCIENCE- XII

CHIEF PATRON:

SH. SANTOSH KUMAR MALL, IAS
(COMMISSIONER, KVS)

PATRON:

MR. C. NEELAP
(DEPUTY COMMISSIONER, GUWAHATI REGION)

GUIDE:

Sh. J. PRASAD, ASSISTANT COMMISSIONER, GUWAHATI REGION
Sh. D. PATLE, ASSISTANT COMMISSIONER, GUWAHATI REGION
Dr. S. BOSE, ASSISTANT COMMISSIONER, GUWAHTI REGION

COORDINATOR:

MR. VISHNU DUTT TAILOR
PRINCIPAL KV CRPF (GC) AMERIGOG GHY

SUBJECT CONTRIBUTORS:-

1. Mr. VIJAY KUMAR, PGT (Comp. Sc.) K.V. CRPF AMERIGOG GHY
2. Dr. K K MOTLA, PGT (Comp. Sc.) K.V. NFR MALIGAON GHY

How to use Study Material:

- It will be much beneficial to those students who need special care and attention. I am sure, thorough study and practicing similar patterns of questions of this material will help such students secure 60% and above.
- However it is not 100% substitute for Textbook.
- Minimum a set of five questions with answers from each questions of the Board Pattern Question Paper are included, keeping in mind the needs and interest of target group.
- Concepts in every unit have been explained using notes / solutions to questions / guidelines in a simple language.
- Practice and peer group discussion on this study material will definitely improve the confidence level of the students when they solve the questions.
- Now you are welcome to the ... content ...

Weightage to different topics/content units

S. No	UNIT	VSA 1 Mark	SA I 2 Marks	SA II 3 Marks	LA 4 Marks	Total
1	Review of C++ covered in Class XI	1 (1)	8 (4)	3 (1)		12 (6)
2	Object Oriented Programming in C++					
	a) Introduction to OOP using C++		2 (1)		4 (1)	6 (2)
	b) Constructor & Destructor		2 (1)			2 (1)
	c) Inheritance				4 (1)	4 (1)
3	Data Structure & Pointers					
	a) Address Calculation			3 (1)		3 (1)
	b) Static Allocation of Objects		2 (1)	3 (1)		5 (2)
	c) Dynamic Allocation of Objects				4 (1)	4 (1)
	d) Infix & Postfix Expressions		2 (1)			2 (1)
4	Data File Handling in C++					
	a) Fundamentals of File Handling	1 (1)				1 (1)
	b) Text File		2 (1)			2 (1)
	c) Binary Files			3(1)		3 (1)
5	Databases and SQL					
	a) Database Concepts		2 (1)			2 (1)
	b) Structured Query Language		2 (1)		4 (1)	6 (2)
6	Boolean Algebra					
	a) Introduction to Boolean Algebra & Laws		2 (1)			2 (1)
	b) SOP & POS	1 (1)				1 (1)
	c) Karnaugh Map			3 (1)		3 (1)
	d) Basic Logic Gates		2 (1)			2 (1)
7	Communication & Open Source Concepts					
	a) Introduction to Networking	2 (2)				2 (2)
	b) Media, Devices, Topologies & Protocols				4 (1)	4 (1)
	c) Security	2 (2)				2 (2)
	d) Webservers	1 (1)				1 (1)
	e) Open Source Terminologies	1 (1)				1 (1)
TOTAL		9 (9)	26 (13)	15 (5)	20 (5)	70 (32)

Review of C++ covered in Class XI

Questions based on Header Files

Very Short Answer Questions (1 mark)

Q1. Write the names of the header files to which the following belong:

(i) isdigit() (ii) strcmp() (iii) fabs() (iv) gets() (v) eof()
(vi) setw() (vii) exit() (viii) tolower() (ix) ceil() (x) feof()
(xi) strupr() (xii) atoi() (xiii) setprecision() (xiv) floor() (xv) remove()
(xvi) strstr() (xvii) put() (xviii) puts() (xix) exp() (xx) free()
(xxi) fwrite() (xxii) write() (xxiii) setiosflags() (xxiv) sin() (xxv) abs()

Ans : (i) ctype.h (ii) string.h (iii) math.h (iv) stdio.h (v) iostream.h
(vi) iomanip.h (vii) process.h (viii) ctype.h (ix) math.h (x) stdio.h
(xi) string.h (xii) stdlib.h (xiii) iomanip.h (xiv) math.h (xv) stdio.h
(xvi) string.h (xvii) iostream.h (xviii) stdio.h (xix) math.h (xx) stdlib.h
(xxi) stdio.h (xxii) iostream.h (xxiii) iomanip.h (xxiv) math.h (xxv) stdlib.h / math.h

Q2. Name the header file(s) that shall be needed for successful compilation of the following C++ code:

```
void main( )
{
        char subject[30];
        strcpy(subject, ”Computer Science”);
        puts(subject);
}
```

Ans : string.h
stdio.h

Note: Marks are not to be deducted if any additional header file is mentioned. But these header files must be written.

Q3. Name the header file(s) that shall be needed for successful compilation of the following C++ code:

```
void main( )
{
        char name[20];
        gets(name);
        cout<<setw(20)<<name;
}
```

Ans : iomanip.h
stdio.h

Q4. Name the header file(s) that shall be needed for successful compilation of the following C++ code:

```
void main( )
{
        char a, b;
        a = getchar();
        b = toupper(a)
        cout<<”\nThe uppercase character of “<< a <<” is “ <<b;
}
```

Ans : ctype.h
stdio.h

Concept Questions based on C++ Review (2 marks)

Q1. What is the difference between a keyword and an identifier in C++? Give examples of both.

Ans : Keyword is a special word that has a special meaning and purpose. Keywords are reserved and are few. For example: goto, for, while, if, else etc.

Identifiers are the user-defined name given to a part of a program. Identifiers are not reserved. It should be the name of any keyword. For example: name, stud, _myfile, op etc.

Q2. What is a reference variable? What is its usage?

Ans : A reference variable is an alias name for a previously defined variable. The usage of it is that the same data object can be referred to by two names and these names can be used interchangeably.

Q3. Write two advantages of using include compiler directives.

Ans: (i) The #include compiler directive lets us include desired header files in our program which enables us work with all declaration / definitions / macros inside the included header file(s).

(ii) It supports modularity.

Q4. Differentiate between a Logical Error and Syntax Error. Also give suitable examples of each in C++.

Ans : Logical Error: Error occurred due to incorrect logic applied by the programmer.

Syntax Error: Error occurred due to not following the proper grammar/syntax of the language OR the error occurred due to violating rules of the programming language

Example:

```
//Program to find area and perimeter of rectangle
void main()
{
        int A=10, B=20, AR, P;
        AR=2*(A*B);                //Logical Error – Wrong Formula
        P=2*(A+B);
        cout<<A<<P >>endl;         //Syntax Error – Use of >> with cout
}
```

Q5. What is the difference between Global Variable and Local Variable?

Ans:

Global Variable	Local Variable
• It is a variable, which is declared outside all the functions • It is accessible throughout the program.	• It is a variable, which is declared with in a function or with in a compound statement • It is accessible only within a function/ compound statement in which it is declared.

```
#include<iostream.h>
float NUM=900;              //NUM is a global variable
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); }
```

Q6. What is the difference between Object Oriented Programming and Procedural Programming?
Ans :

Object Oriented Programming	Procedural Programming
• Emphasis on Data • Follows Bottom-Up approach in program design • Data hiding feature prevents accidental change in data • Features like data encapsulation, polymorphism, inheritance are present	• Emphasis on doing things (functions) • Follows Top-down approach in program design • Presence of Global variables increase chances of accidental change in data • Such features are not available

Q7. Differentiate between a Call by Value and Call by Reference, giving suitable examples of each?
Ans:

Call by Value	Call by Reference
• The called function creates its own copies of the original values sent to it. • Any changes that are made in the function run, changes in the original values are not reflected.	• The called function accesses and works with the original values using their references. • Any changes that occur in the function run, changes in the original values are reflected.

void change(int b)
{
b = 10;
}
void main()
{
int a = 5;
cout<<”\n a = “<<a;
change(a);
cout<<”\n a = “<<a;
}
Output will be: a = 5
a = 5
	void change(int &b)
	{
	b = 10;
	}
	void main()
	{
	int a = 5;
	cout<<”\n a = “<<a;
	change(a);
	cout<<”\n a = “<<a;
	}
	Output will be: a = 5
	a = 10

Q8. What is a parameter? Differentiate between an actual and a formal parameter with an example?
Ans : Parameter is the variable / value passed to a function or the variable that is used as the incoming values in a function. The variables / values passed to a function are called actual parameters. The variables that are used as the incoming values in a function are called formal parameters. For Example:

```
void change(int b)          // b is the formal parameter
{
        b = 10;
}
void main()
{
        int a = 5;
        change(a);          // a is the actual
        parameter  cout<<”\n a = “<<a;
}
```

Q9. Enlist any four jump statements with their uses.
Ans : (i) goto : A goto statement can transfer the program control anywhere in the program.

(ii) break : A break statement enables a program to terminate of the loop/block, skipping any code in between.

(iii) continue : A break statement enables a program to force the next iteration to take place, skipping any code in between.

(iv) return : A return statement is used to return from a function.

Q10. How are the following related to one another?

(i) array and structure (ii) structure and class

Ans: (i) Array is a group of items of the same data types whereas structure brings together a group of related data items of any data types.

(ii) Structure is actually a class (in C++) declared with keyword struct. By default, all members are public in a structure; on the other hand all members are private by default in a class.

<u>Questions based on Program Errors</u> (2 marks)

Note : Errors are explained in the solutions. While solving such type of question, a student should carefully go through correctness of each statement and the logic of the whole program. Students are advised to practice various questions to develop the skill of finding the errors.

Q1. Rewrite the following program after removing the syntactical errors (if any). Underline each correction.

```
#include<iostream.h>
int func(int y =10, int &x)
{
        if(x%y = 0) return ++x ; else return y-- ;
}
void main()
{
        int p = 20, q =
        23; r =
        func(p,q);
        cout>>p>>q>>r
        ;
}
```

Ans:

```
#include<iostream.h>
int func(int y , int &x)          // violating the rule of Default argument
{
        if(x%y = = 0) return ++x ; else return y-- ;          // = = relational operator
}
void main()
{
        int p = 20, q = 23;
       int r = func(p,q);                  // r should be declared
       cout << p << q << r;                // << operator for cout
}
```

Q2. . Rewrite the following program after removing the syntactical errors (if any). Underline each correction.

```
#include<iostream.h>
void main()
{
        int X[ ]={60,50,30,40},Y; count=4;
        cin>>Y;
        for(i=count-1;i>=0;i--)
        switch(i)
```

```
                {
                        case 1;
                        case 2: cout<<Y * X; break;
                        case 3: cout<<Y+ Y;
                }
        }
```

Ans:

```
#include<iostream.h>
void main()
{
        int X[ ]={60,50,30,40},Y, count=4;         // multiple declaration separated by comma
        cin>>Y;
        for(int i=count-1; i>=0; i--)              // i should be declared
        switch(i)
        {
                case 1:;                            // case should follow by :
                case 2: cout<<Y*X[i]; break;        // Lvalue required for X
                case 3: cout<<Y + Y;
        }
}
```

Q4. Rewrite the following program after removing the syntactical errors (if any). Underline each correction.

```
 #include <iostream.h>
struct Pixels
{ int Color,Style;}
void ShowPoint(Pixels P)
{ cout<<P.Color,P.Style<<endl;}
void main()
{
        Pixels Point1=(5,3);
        ShowPoint(Point1);
        Pixels Point2=Point1;
        Color.Point1+=2;
        ShowPoint(Point2);
}
```

Ans:

```
#include <iostream.h>
struct Pixels
{       int Color,Style;} ;             // Definition of structure Pixels must be ended with ;
void ShowPoint(Pixels P)
{       cout<<P.Color << P.Style<<endl;}        // In cascading of cout, << to be used
void main()
{
        Pixels Point1 = {5,3};          // { } to be used to initialise of members of the object
        ShowPoint(Point1);
        Pixels Point2=Point1;
        Point1.Color+=2;                // member to followed by the object using . operator
        ShowPoint(Point2);
}
```

Questions based on Finding Outputs using random()

Short Answer Questions (2 marks)

Note: random(n) generates the numbers randomly from 0 to n–1. For example: random(20) generates randomly from 0 to 19. Explanations are given at the end of each solution.

Q1. In the following program, if the value of N given by the user is 50, what maximum and minimum values the program could possibly display?

```
#include <iostream.h>
#include <stdlib.h>
void main()
{
        int N,Guessme;
        randomize();
        cin>>N;
        Guessme=random(N) + 5;
        cout<<Guessme<<endl;
}
```

Ans : Minimum : 5 Maximum : 54

Explanation : Since random(50) gives a number randomly from 0 to 49. If it returns 0 i.e. minimum for random(50), the minimum value for Guessme will be 0 + 5 = 5. If it returns 49 i.e. maximum for random(50), the maximum value for Guessme will be 49 + 5 = 54.

Q2. Study the following program and select the possible output from it:

```
#include<iostream.h>
#include<stdlib.h
> const int
Max=3; void
main( )
{
        randomize();
        int Number;
        Number=50 + random(Max);
        for(int P=Number; P >=50; P - -)
              cout<<P<<"#";
        cout<<endl;
}
```

(i) 53#52#51#50#

(ii) 50#51#52#

(iii) 50#51#

(iv) 51#50#

Ans: 51#50#

Explanation : Since random(3) gives a number randomly from 0 to 2. So, the values of Number ranges from 50 (50+0) to 52 (50+2) and the Number should be displayed in descending order according to the programme. 53 is not possible, so the answer is (iv).

Q3. In the following program, if the value of N given by the user is 20, what maximum and minimum values the program could possibly display?

```
#include <iostream.h>
#include <stdlib.h>
void main()
{
        int N,Guessnum;
        randomize();
        cin>>N;
        Guessnum= random(N – 10)+10;
        cout<<Guessnum<<endl;
}
```

Ans : Maximum Value: 19 Minimum Value: 10

Q4. In the following program, if the value of Guess entered by the user is 65, what will be the expected output(s) from the following options (i), (ii), (iii) and (iv)?

```
#include <iostream.h>
#include <stdlib.h>
void main()
{
        int Guess;
        randomize();
        cin>>Guess;
        for (int I=1;I<=4;I++)
        {       New=Guess+random(I);
                cout<<(char)New;

         }
 }
```

(i) ABBC
(ii) ACBA
(iii) BCDA
(iv) CABD

Explanation : In first iteration I = 1 and random(I) gives 0. So, New = 65 + 0, hence output is A. In first iteration I = 1, So, New = 65 + 0, hence output is A. In second iteration I = 2 and random(I) gives 0 or 1. So, New = 65 or 66, hence output is A or B. In third iteration I=3 and random(I) gives 0 to 2. So, New = 65 or 66 or 67, hence output is A or B or C. In fourth iteration I=4 and random(I) gives 0 to 3. So, New = 65 or 66 or 67 or 68, hence output is A or B or C or D. Option (i) satisfies all the above only, so it is the answer.

Ans: (i) ABBC

Questions based on Finding Outputs

Short Answer Questions (2 marks)

Note : While solving such type of question, a student should carefully go through the logic of the whole program and concepts used in it. Various patterns of questions are given below. Students are advised to practice various questions to develop the skill of finding the output.

Q1. Find the output of the following program:

```
#include <iostream.h>
struct Game
{
        char Magic[20];int Score;
};
void main()
{
        Game M={“Tiger”,500};
        char *Choice;
        Choice=M.Magic;
        Choice[4]=’P’;
        Choice[2]=’L’;
        M.Score+=50;
        cout<<M.Magic<<M.Score<<endl;
        Game N=M;
        N.Magic[0]=’A’;N.Magic[3]=’J’;
        N.Score-=120;
        cout<<N.Magic<<N.Score<<endl;
}
```

Ans: **TiLeP550**
AiLJP430

Q2. Find the output of the following program:

```
#include<iostream.h>
void main()
{
        int Numbers[] = {2,4,8,10};
        int *ptr = Numbers;
        for (int C = 0; C<3; C++)
        {
                cout<< *ptr << “@”;
                ptr++;
        }
        cout<<endl;
        for(C = 0; C<4; C++)
        {
                (*ptr)*=2;
                --ptr;
        }
        for(C = 0; C<4; C++)
                cout<< Numbers [C]<< “#”;
        cout<<endl;
}
```

Ans : **2@4@8@**
4#8#16#20#

Q3. Find the output of the following program:

```
#include <iostream.h>
void fun( int p, int *q )
{
         p = *(q) += 2;
}
void main()
{
    int x, b = 5, a[2] = { 10, 20 };
    for( x = 1; x < 3 ; x++)
   {
       fun( a[x – 1], &b);
       cout<< “\n” << a[x - 1] <<” : “ << b;
    }
}
```

Ans: **10 : 7**
20 : 9

Q4. Find the output of the following program:

```
#include <iostream.h>
void main()
{
     long NUM = 98534210;
     int f=0,s=0;
     do
    {
         int rem = NUM % 10;
         if( rem % 2 == 0)
                 f += rem;
         else
                 s += rem;
         NUM /= 10;
      }while(NUM > 0);
     cout<<“\n”<<f<<” – “<<s<<“ = ”<<f – s;
}
```

Ans : **14 – 18 = – 4**

Q5. Find the output of the following program:

```
#include<iostream.h>
int a=10;
void main()
{
        void demo(int &, int , int * );
        int a=20,b=5;
        demo(::a,a,&b);
        cout<<"\n"<<::a<<"*"<<a<<"*"<<b;
}
void demo(int &x, int y, int *z)
{
         a = a + x;
         y = y * a;
        *z = a+y;
        cout<<"\n"<<x<<"*"<<y<<"*"<<*z ;
}
```

Ans: 20 * 400 * 420
20 * 20 * 420

Q6. Find the output of the following program:

```
#include <iostream.h>
#include<ctype.h>
void main()
{
        char *name= "ThE bESt meN wIN";
        for ( int i =0;  name[i]!='\0' ; i++)
       {
                if ( islower( name[i]) )
                       name[i] = toupper(name[i]) ;
                else
                if( isupper(name[i]) )
                   if ( i%2 == 0)
                      name[i] – – ;
                   else
                      name[i] = tolower(name[i–1]);
           }
           cout<<name;
}
```

Ans : SHD BbRT MEe WHh

Questions based on Finding Outputs

Short Answer Questions (3 marks)

Q1. Find the output of the following program:

```
#include <iostream.h>
#include<string.h>
struct KEY
{
            char word[10];
            int count;
};
void changekeyword(KEY somekey);
void main()
{
            KEY aKEY;
            strcpy(aKEY.word, "#define");
            aKEY.count=10;
            cout<<aKEY.word<< "\t"<<aKEY.count<< "\n";
            changekeyword(aKEY);
            cout<<aKEY.word<< "\t"<<aKEY.count<< "\n";
}
void changekeyword(KEY somekey)
{
            strcpy(somekey.word, "const");
            somekey.count += 1;
            cout<<somekey.word<< "\t" <<somekey.count<< "\n";
}
```

Ans : #define 10
#const 11
#define 10

Q2. Find the output of the following program:

```
#include <iostream.h>
int modify(int temp = 2)
{
        if(temp % 3 == 0)
                temp = temp + 1;
        else
                temp = temp + 3;
        return temp;
}
void doupdation(int m, int &n)
{
        static int i;
        i++;
        m = n + i;
        if(n > 10)
                n = modify();
        else
                n = modify(n);
        cout<<m<<" ; "<<n<<endl;
}
void main()
{
        int x = 8, y = 20;
        doupdation(x,y);
        cout<<x<<" ; "<<y<<endl;
        doupdation(y,x);
        cout<<x<<" ; "<<y<<endl;
        doupdation(y,x);
        cout<<x<<" ; "<<y<<endl;
}
```

Ans: **21 ; 5**
8 ; 5
10 ; 11
11 ; 5
14 ; 5
5 ; 5

Q3. Find the output of the following program:

```
#include <iostream.h>
void chang ( int &x, int y)
{
        int temp;
        temp=x;
        x=y;
        y=temp;
        cout<<"\n"<<x<<" : "<<y;
}
void main()
{
        int a=10, b;
        b= a++ + ++a + ++a + a++ + 2;
        cout<<"\n"<< a <<" : "<<b;
        chang( a , b );
        cout<<"\n"<< a++ <<" : "<< ++b;
}
```

Ans : **14 : 50**
50 : 14
50 : 51

Q4. Give the output of the following program:

```
#include<iostream.h> void
main()
{
        int a, *b, **c, ***d;
        a=12, b=&a, c=&b, d=&c; ***d*=5;
        cout<<"\n"<<a<<","<< a + *b; (**c) +=
        10;
        cout<<"\n"<<**c + ***d;
        (***d)+= 10; cout<<"\n"<< a +
        *b;
}
```

Ans: **60 , 120**
140
160

Q5. Give the output of the following program:

```
#include <iostream.h>
#include <string.h>
#include <ctype.h>
void funnystr(char *s, int n = 2)
{
        int i = n;
        while(i < strlen(s))
        {
                s[i] = '-';
                i = i + n;
        }
        i = 0;
        while(s[i] != '\0')
        {
                if(s[i] > 'A' && s[i] < 'P')
                        s[i] = tolower(s[i]);
                else if(s[i] > 'a' && s[i] < 'p')
                {
                        if(i % 3 == 0)
                                s[i] = tolower(s[i-1]);
                        else
                                s[i] = tolower(s[i]);
                }
                i++;
        }
  }
  void main()
  {
        char str[] = "MiCroSoFT";
        funnystr(str,3); cout<<str;
  }
```

Ans : mic–oS–fT

Introduction to OOP using C++

Object Oriented Programming Concepts

Short Answer Questions

Q1. Define object.
Ans. Object is an identifiable entity with some characteristics and behaviour.

Q2. Define class.
Ans. A class is group of objects that share common properties and relationships.

Q3. Define data abstraction.
Ans. Abstraction refers to the act of representing essential features without including the background details or explanations.

Q4. What is inheritance?
Ans. Inheritance is the capabilities of one class of things to inherit capabilities or properties from another class.

Q5. Define modularity.
Ans. Modularity is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules.

Q6. Define encapsulation.
Ans. The wrapping up of data and operations / functions (that operate on the data) into a single unit (called class) is known as encapsulation .

Q7. What is polymorphism?
Ans. Polymorphism is the ability for a message or data to be processed in more than one form.

Q8. How polymorphism is implemented in C++?
Ans. Polymorphism is implemented in C++ through overloaded functions, overloaded operators and virtual functions.

Q9. How inheritance is implemented in C++?
Ans. Inheritance is implemented in C++ by specifying the base class from which the derived class of the base class is defined.

Q10. What is data hiding? How it is implemented in C++?
Ans. Data hiding is a property whereby the internal data structure of an object is hidden from the rest of the program. Data hiding is implemented in C++ by private and protected members of class.

Q11. How data abstraction can be implemented in C++?
Ans. Data abstraction can be implemented in C++ by using public members of class.

Q12. What is programming paradigm?
Ans. A programming paradigm defines the methodology of designing and implementing programs using the key features and building blocks of a programming language.

Q13. What is procedural programming paradigm?
Ans. In procedural programming paradigm, data is shared among all the functions. Its emphasis is on doing things rather than using the data.

Q14. What are the advantages of OOP?

Ans : The advantages of OOP are Reusability of code, Ease of comprehension, ease of fabrication and maintenance, ease of redesign and extension.

Q15. What do you understand by functional overloading? Give an example illustrating its use in a C++ program.

Ans: A function name having several definitions that are differentiable by the number or types of their arguments is known as functional overloading. For example:

```
float volume(float a )
{ return a * a * a; }
float volume(float a, float b )
{ return a * a * b; }
float volume(float a, float b, float c)
{ return a * b * c; }
```

CLASSES AND OBJECTS

SOME MPORTANT QUESTIONS TO REFRESH THE CONCEPT

Q1. Write four attributes associated with declaration of classes.

Ans. The attributes associated with declaration of classes are Data Members, Member Functions, Program access levels and Class Tagname.

Q2. Define data members and member functions.

Ans. Data members are exactly like the variables in a structure. They are normally made Member function is function defined within a class that acts on the data members in the class. They are normally made public.

Q3. Write the scope rules of members of a class.

Ans. Scope rules of members of a class are:-

Private	These members can be accessed only by the member functions & friend functions of the class. It can not be inherited.
Protected	These members can be accessed only by the member functions & friend functions of the class. It can be inherited.
Public	These members can be directly accessed by any function.

Q4. How many types of functions are used in a class?

Ans. 1. Accessor function 2. Mutator functions 3. Manager functions

Q**5.** When will you make a function inline?

Ans. A function is made inline when it is very small and does not return any value. Also it does not contain any loop and static variables.

Q6. What is scope resolution operator?

Ans. This operator(::) is used in situations where a global variable exists with the same name as a local variable. This is also used when the member functions are declared outside the class definition; the function name is preceded by the class name and the binary scope resolution operator(::).

Q7. Define static data members.

Ans. Static data members are initialized to zero when the first object of its class is created. No other initialization is permitted. Only one copy of that member is created for the entire class and is shared by all the objects for that class.

Q8. How is memory allocated to a class and its objects?

Ans. When a class is defined, memory is allocated for its member functions and they are stored in the memory. When object is created, separate memory space is allocated for its data members. All objects work with the same copy of member function shared by all.

BOARD PATTERN QUESTIONS:

Long Answer Questions (4 marks)

Q1. Define a class **employee** with the following specifications;

Private members of class **employee :**

empno	integer
ename	20 characters
basic,hra,da	float
netpay	float
calculate( )	A function to calculate basic + hra + da with float return type

Public member functions of class **employee :**

havedata( )	function to accept values for empno, sname, basic , hra ,da and invoke
calculate( )	to calculate netpay
dispdata( )	function to display all the data members on the screen .

Ans.

```
class employee
{
    int empno;
    char ename[20];
    float basic,da,hra;
    float netpay;
    float calculate( )
    {
        return (basic + da + hra);
     }
  public :
        void havedata( )
        {
            cout << ”Enter employee no”;
            cin >> empno;
            cout << ”\n Enter name “;
            cin >> ename;
            cout << ”\n Enter basic salary”;
            cin >> basic;
            cout << ”\n Enter DA “;
            cin >> da;
            cout << “\n Enter HRA “;
            cin >> hra;
            netpay = calculate( );
        }
         void dispdata( )
        {
             cout << “\n Employee no is “<< empno << endl ;
             cout << “\n Name is “ << ename << endl ;
             cout << “\n Basic Salary  is” << basic << endl ;
             cout << “\n DA is “ << endl ;
             cout << “\n HRA is “ << hra << endl ;
             cout << “\n Netpay is “ << netpay << endl ;
        }
} ;
```

Q2. Define a class **Student** for the following specifications :

Private members of the **Student** are :

roll _ no integer

name array of characters of size 20

class _ st array of characters of size 8
marks array of integers of size 5
percentage float
calculate that calculates the percentage marks

Public members of the **Student** are :

readmarks reads mark and invoke the calculate function
displaymarks prints the data.

Ans.
```
class Student
{
    int roll _ no ;
    char name[20] ;
    char class _ st[8] ;
    int marks[5] ;
    float percentage ;
    void calculate( ) ;
    public :
            void readmarks( ) ;
            void displaymarks( ) ;
} ;
void Student : : calculate( )
{
     float total = 0 ;
     for ( int I =0 ; I < 5; I ++ )
           total + = marks[I] ;
     percentage = total / 5 ;
}
void  Student : : readmarks( )
{
      cout << " \n Enter 5 marks : " ;
      for ( int I =0 ; I < 5 ; I ++ )
           cin >> marks[I] ;
      calculate( ) ;
}
void  Student : : displaymarks( )
{
      cout << "\n The 5 marks are : " ;
      for ( int I = 0 ; I < 5 ; I ++ )
            cout << marks[I] << endl ;
      cout << " \n Percentage = " << percentage << endl ;
 }
```

Q3. Define a class **DONOR** with the following specifications :

Private :

Donor number	integer
Name	20 characters
Blood group	2 characters

Public :

Input( )	A function to accept all the information
Output( )	A function to display all the information
Checkgroup( )	A function with char * return to return Blood Group

Define both the number functions with their given description.

Ans.
```
class DONOR
{   int d_no ;
    char name[20] ;
    char b_grp[2] ;
public :
       void input( )
       {
            cout << "\n Enter the details as donor no, name and blood group : " ;
             cin >> d_no ; gets ( name ); gets ( b_grp ) ;
       }
      void output( )
      {     cout << "\n The details are " ;
             cout << "\n Name : " << name << "\n Donor No : << d_no ;
             cout << "\n  Blood Group : " << b_grp ;
       }
        char * checkgroup ( )
       {     return ( b_grp ) ;
        }
} ;
```

Q4. Define a class TEST in C++ with following description:

Private Members

- TestCode of type integer
- Description of type string
- NoCandidate of type integer
- CenterReqd (number of centers required) of type integer
- A member function CALCNTR() to calculate and return the number of centers as (NoCandidates/100+1)

Public Members

- A function SCHEDULE() to allow user to enter values for TestCode, Description, NoCandidate & call function CALCNTR() to calculate the number of Centres
- A function DISPTEST() to allow user to view the content of all the data members

Ans:

```
class TEST
{
            int TestCode;
            char Description[20];
            int NoCandidate,CenterReqd;
            void CALCNTR();
public:
            void SCHEDULE();
            void DISPTEST();
};
void TEST::CALCNTR()
{           CenterReqd=NoCandidate/100 + 1; }
void TEST::SCHEDULE()
{
            cout<<"Test Code   :";cin>>TestCode;
            cout<<"Description :";gets(Description);
            cout<<"Number      :";cin>>NoCandidate;
            CALCNTR();
}
void TEST::DISPTEST()
{
            cout<<"Test Code   :"<<TestCode<<endl;
            cout<<"Description :"<<Description<<endl;
            cout<<"Number      :"<<NoCandidate<<endl;;
            cout<<"Centres     :"<<CenterReqd<<endl;;
}
```

CONSTRUCTOR AND DESTRUCTOR

Q1. What is constructor?

Ans. A member function with the same name as its class is called constructor and it is used to initialize the objects of that class type with a legal initial value.

Q2. What is destructor?

Ans. A destructor is a member function having same name as its class but preceded by ~ sign and it deinitialises an object before it goes out of scope.

Q3. What are different types of constructors?

Ans. The different types of constructors are Default constructor, Parameterized constructor and Copy constructor.

Q4. What is default constructor?

Ans. A constructor that accepts no parameter is called the default constructor.

Q5. What is parameterized constructor?

Ans. A constructor that accepts parameters for its invocation is known as parameterized constructor.

Q6. What is copy constructor?

Ans. A copy constructor is a constructor that defines and initializes an object with another object. It takes the form classname(classname). For a copy constructor there must be a default constructor or a parameterized constructor.

BOARD PATTERN QUESTIONS - Short Answer Questions (2 marks)

✪Note : Two options with the following pattern are generally asked in the Board Exam. Various optional questions are given below.

Q7. Answer the following questions after going through the following class:

```
class Seminar
{
        int Time;
public:
        Seminar();                                          //Function 1
        void Lecture()                                      //Function 2
        {cout<<"Lectures in the seminar on"<<end1;}
        Seminar(int);                                       //Function 3
        Seminar(Seminar &abc);                              //Function 4
        ~Seminar()                                          //Function 5
        {  cout<<"Vote of thanks"<<end1;
};
```

(i) In Object Oriented Programming, what is Function 5 referred as and when does it get invoked/called?

Ans : Function 5 is referred as destructor and it is invoked as soon as the scope of the object gets over.

(ii) In Object Oriented Programming, which concept is illustrated by Function 1, Function 3 and Function 4 all together?

Ans : Constructor Overloading (Polymorphism)

(iii) Which category of constructor - Function 1 belongs to? Write an example illustrating the calls for Function 1.

Ans : Default Constructor. Example to invoke function 1 → Seminar S;

(iv) Which category of constructor - Function 3 belongs to? Write an example illustrating the calls for Function 3.

Ans : Parameterised Constructor. Example to invoke function 3 → Seminar A(8);

(v) Which category of constructor - Function 4 belongs to? Write an example illustrating the calls for Function 4.

Ans : Copy Constructor. Example to invoke function 4 → Seminar S2(S);
Or Seminar S2 = S;

(vi) Write an example illustrating the calls for Function 3 explicitly.

Ans : Seminar A = Seminar(8);

(vii) Write an example illustrating the calls for Function 4 explicitly.

Ans : Seminar S2 = Seminar(S);

(viii) Write the complete definition for Function 1 to initialize Time as 30.

Ans : Seminar :: Seminar()
{ Time = 30; }

(ix) Write the complete definition for Function 3 to initialize Time with Mytime as parameter to the Function 3.

Ans : Seminar :: Seminar(int Mytime)
{ Time = Mytime; }

(x) Write the complete definition for Function 4.

Ans : Seminar :: Seminar(Seminar &abc)
{ Time = abc.Time; }

Q8. Answer the following questions after going through the following class:

```
class Complex
{
        int x;
        int y;
public:
        Complex();                                              //Function 1
        void disp()                                             //Function 2
        {cout<<”The Complex number is : “<<x<<” + “<<y<<”i”<<end1;}
        Complex(int, int);                                      //Function 3
        Complex(Complex &abc);                                  //Function 4
};
```

(i) Which category of constructor - Function 1 belongs to? Write an example illustrating the calls for Function 1.

Ans : Default Constructor. Example to invoke function 1 → Complex C;

(ii) Which category of constructor - Function 3 belongs to? Write an example illustrating the calls for Function 3.

Ans: Parameterised Constructor. Example to invoke function 3→Complex C(6,8);

(iii) Which category of constructor - Function 4 belongs to? Write an example illustrating the calls for Function 4.

Ans : Copy Constructor. Example to invoke function 4 → Complex C2(C);
Or Complex C2 = C;

(iv) Write an example illustrating the calls for Function 3 explicitly.

Ans : Complex C = Complex(6,8);

(v) Write an example illustrating the calls for Function 4 explicitly.

Ans : Complex C2 = Complex(C);

(vi) Write the complete definition for Function 1 to initialize x as 10 and y as 20.

Ans : Complex :: Complex ()

```
{ x = 10 ; y = 20; }
```

(vii) Write the complete definition for Function 3 to initialize the data members with p and q as parameters to the Function 3.

Ans :
```
Complex :: Complex(int p, int q)
        { x = p; y = q; }
```

(viii) Write the complete definition for Function 4.

Ans :
```
Complex :: Complex (Complex &abc)
        { x = abc.x;
          y = abc.y;
        }
```

Inheritance

SOME MPORTANT QUESTIONS TO REFRESH THE CONCEPT

Q1. Write the reasons behind the introduction of inheritance in OOP.

Ans. The major reasons behind the introduction of inheritance in OOP are:(i) It ensures the closeness with the real world models, (ii) idea of reusability, (iii) transitive nature of inheritance.

Q2. What are the different forms of inheritance?

Ans. The different forms of inheritance are (i) Single Inheritance, (ii) Multiple Inheritance, (iii) Hierarchical Inheritance, (iv) Multilevel Inheritance, (v) Hybrid Inheritance.

Q3. How does the access of inherited members depend upon their access specifiers and the visibility modes of the base class?

Ans.

Access specifier / Visibility mode	public inheritance	protected inheritance	private inheritance
public member in base class	**public** in derived class	**protected** in derived class	**private** in derived class
protected member in base class	**protected** in derived class	**protected** in derived class	**private** in derived class
private member in base class	hidden in derived class	hidden in derived class	hidden in derived class

Q4. Write the different ways of accessibility of base class members.

Ans.

Access Specifier	Accessible from own class	Accessible from derived class (Inheritable)	Accessible from objects outside class
public	Yes	Yes	Yes
protected	Yes	Yes	No
private	Yes	No	No

Q5. How is the size of a derived class object calculated?

Ans. The size of a derived class object is equal to the sum of sizes of data members in base class and the derived class.

Q6. In what order are class constructors and class destructors called when a derived class object is created or destroyed?

Ans. When the object of a derived class is created, firstly the constructor of the base class is invoked and then, the constructor of the derived class is invoked.

On the other hand, when the object of a derived class is destroyed, firstly the destructor of the derived class is invoked and then, the destructor of the base class is invoked.

Note : Observe the following questions and their solutions. The above concepts are implemented. Practice of such questions will definitely clear your doubts and improve your confidence level. Discussion with your friends and distinguish the varieties of these questions.

BOARD PATTERN QUESTIONS Long Answer Questions (4 Marks)

Q1. Answer the questions (i) to (iv) based on the following:

```
class vehicle
 {
                  int wheels;
  protected:
                  int passenger;
 public:
     void inputdata( );
     void outputdata( );
 };
```

```
class heavyvehicle : protected vehicle
 {
                  int diesel_petrol;
  protected:
                  int load;
  public:
      void readdata(int, int);
      void writedata( );
 };
```

```
class bus : private heavyvehicle
 {
                  char make[20];
   public:
          void fetchdata( );
          void displaydata( );
 };
```

i) Write the member(s) that can be accessed from the object of bus.
ii) Write the data member(s) that can be accessed from the function displaydata( ).
iii) How many bytes are required by an object of bus and heavyvehicle classes respectively?
iv) Is the member function outputdata( ) accessible to the objects of the class heavyvehicle?

Ans : (i) fetchdata(), displaydata() (ii) make, load, passanger
(iii) for the object of bus – 28 bytes, for the object of heavyvehicle – 8 bytes (iv) No

Q2. Answer the questions (i) to (iv) based on the following:

```
class livingbeing
{
            char specification[20];
            int averageage;
   public:
            void read();
            void show();
};
```

```
class ape : private livingbeing
{
            int no_of_organs;
int no_of_bones;
   protected:
            int iq_level;
   public:
            void readape();
            void showape();
};
```

```
class human : public ape
{
            char race[20];
            char habitation[30];
   public:
            void readhuman();
            void showhuman();
};
```

(i) Write the members which can be accessed from the member functions of class human.
(ii) Write the members, which can be accessed by an object of class human.
(iii) What is the size of an object (in bytes) of class human?
(iv) Write the class(es) which objects can access read() declared in livingbeing class.

Ans : (i) race, habitation, iq_level, readhuman(), showhuman(), readape(), showape()
(ii) readhuman(), showhuman(), readape(), showape()
(iii) 78 bytes (iv) livingbeing

Q3. Answer the questions (i) to (iv) based on the following:

```
class parent
{
            char name[20];
protected:
            int son;
public:
            void inputdata();
            void outputdata();
};
```

```
class father : protected parent
{
            int daughter;
protected:
            int baby;
public:
            void readdata();
            void writedata();
};
```

```
class mother : public father
{
            int child;
public:
            void fetchdata();
            void dispdata();
};
```

(i) In case of the class father, what is the base class of father and what is the derived class of father?
(ii) Write the data member(s) that can be accessed from function dispdata().
(iii) Write the member function(s), which can be accessed by an object of mother class.
(iv) Is the member function outputdata() accessible to the objects of father class?

Ans : (i) base class of father – parent, derived class of father – mother
(ii) child, baby, son
(iii) fetchdata(), dispdata(), readdata(), writedata() (iv) No

Data Structure & Pointers

Address Calculation

Short Answer Questions (3 marks)

Formulae of Row Major & Column Major are used in the given questions. Kindly go through it.

Q1. An array x[30][10] is stored in the memory with each element requiring 4 bytes of storage. If the base address of x is 4500, find out memory locations of x[12][8] and x[2][4], if the content is stored along the row.

Ans : Given, W = 4, N = 30, M = 10, Base(x) = 4500

Row Major Formula:

Loc(x[12][8]) = Base(x)+W*(M*I+J)
= 4500 + 4 * (10 * 12 + 8)
= 4500 + 4 * 128
= 4500 + 512
= 5012

Loc(x[2][4]) = Base(x)+W*(M*I+J)
= 4500+ 4*(10*2 + 4)
= 4500 + 4 * 24
= 4500 + 96
= 4596

Q2. An array P[20][30] is stored in the memory along the column with each of the element occupying 4 bytes, find out the Base Address of the array, if an element P[2][20] is stored at the memory location 5000.

Ans : Given, W=4, N=20, M=30, Loc(P[2][20])=5000

Column Major Formula:

Loc(P[I][J]) =Base(P)+W*(N*J+I)
Loc(P[2][20]) =Base(P)+4*(20*20+2)
Base(P) =5000 – 4*(400+2)
=5000 – 1608
=3392

Q3. An array ARR[5][25] is stored in the memory with each element occupying 4 bytes of space. Assuming the base address of ARR to be 1000, compute the address of ARR[5][7], when the array is stored as : (i) Row wise (ii) Column wise.

Ans:

(i) Row wise : Given, W = 4, N = 5, M = 25, Base(ARR) = 1000 Row Major Formula: Loc(ARR[5][7])= Base(ARR)+W*(M*I+J) = 1000 + 4*(25*5 + 7) = 1000 + 4 * 132 = 1000 + 528 = 1528	(ii) Column wise : Given, W = 4, N = 5, M = 25, Base(ARR) = 1000 Column Major Formula: Loc(ARR[5][7])= Base(ARR)+W*(N*J+I) = 1000 + 4*(5*7 + 5) = 1000 + 4 * 40 = 1000 + 160 = 1160

Static Allocation of Objects

Short Answer Questions (2 marks)

Note : Practice the way to write the function definition where array and its size are passed as arguments. Practice with the concepts of accessing the elements of the one / two dimensional array. Apply the suitable logic to solve the given problem and write the coding of it.

Q1. Write a function in C++ to find sum of rows from a two dimensional array. Ans : void MatAdd(int A[ ][ ],int N, int M)

```
{
        for (int R=0;R<N;R++)
        {
                int SumR=0;
                for (int C=0;C<M;C++)
                        SumR+=A[C][R];
                cout<<SumR<<endl;
        }
}
```

Q2. Write a function in C++ to find the sum of both left and right diagonal elements from a two dimensional array (matrix).

Ans : void DiagSum(int A[ ][ ], int N)

```
{
        int
        SumD1=0,SumD2=0;
        for (int I=0;I<N;I++)
        {
              SumD1+=A[I][I];
              SumD2+=A[N-I-1][I];
        }
        cout<<"Sum of Diagonal 1:"<<SumD1<<endl;
        cout<<"Sum of Diagonal 2:"<<SumD2<<endl;
}
```

Q3. Write a function in C++ which accepts an integer array and its size as arguments and replaces elements having even values with its half and elements having odd values with twice its value. eg:

if the array contains : 3, 4, 5, 16, 9

then the function should be rearranged as 6, 2,10,8, 18

Ans: void calc(int x[ ], int m)

```
{
        for(int i=0; i<m; ++i)
        {
                if(x[i]%2==0)
                        x[i]=x[i]/2;
                else
                        x[i]=x[i]*2;
        }
}
```

Q4. Write a user defined function in C++ which intakes one dimensional array and size of array as argument and display the elements which are prime.

If 1D array is 10 , 2 , 3 , 4 , 5 , 16 , 17 , 23

Then prime numbers in above array are: 2 , 3 , 5, 17, 23

Ans: void primedisp(int x[ ], int m)

```
{
```

```
                int nf;
                for(int i=0; i<m; ++i)
                {
                        nf=0;
                        for(int j=1; j<=x[i]; j++)
                                if(x[i] % j == 0)
                                        nf++;
                        if(nf == 2)
                                cout<<"\n"<<x[i];
                }
        }
```

Q5. Write a function in C++ which accepts an integer array and its size as arguments/parameters and exchanges the values at alternate locations.

example : if the array is 8,10,1,3,17,90,13,60 then rearrange the array as 10,8,3,1,90,17,60,13

```
Ans : void exchange(int x[ ], int m)
        {
                int y;
                for(int i=0; i<m; i= i+2)
                {
                        if(m-i == 1)                    // if m (no. of element ) is odd last element
                                i = m;                  // should not alter its position.
                        else
                        {
                                y= x[i];
                                x[i] = x[i+1];
                                x[i+1] = y;
                        }
                }
        }
```

Short Answer Questions (3 marks)

Q1. Write a user defined function DispTen(int A[][4],int N,int M) In c++ to find and display all the numbers, which are divisible by 10. For example if the content of array is:

12	20	13
2	10	30

The output should be

20 10 30

```
Ans:    void DispTen(int A[ ][4], int N, int M)
        {
        int i, j ;
        for(i=0 ; i<N ; i++)
        {
        for(j=0 ; i<M ; j++)
        {
        if(A[i][j]%10= =0)
        cout<<A[i][j]<<" " ;
        } } }
```

Q2 Write a function ALTERNATE (int A[ ] [3], int N, int M) in C++ to display all alternate elements from two-dimensional array A (staring from A [0] [0]). For example:

If the array is containing:

23 54 76

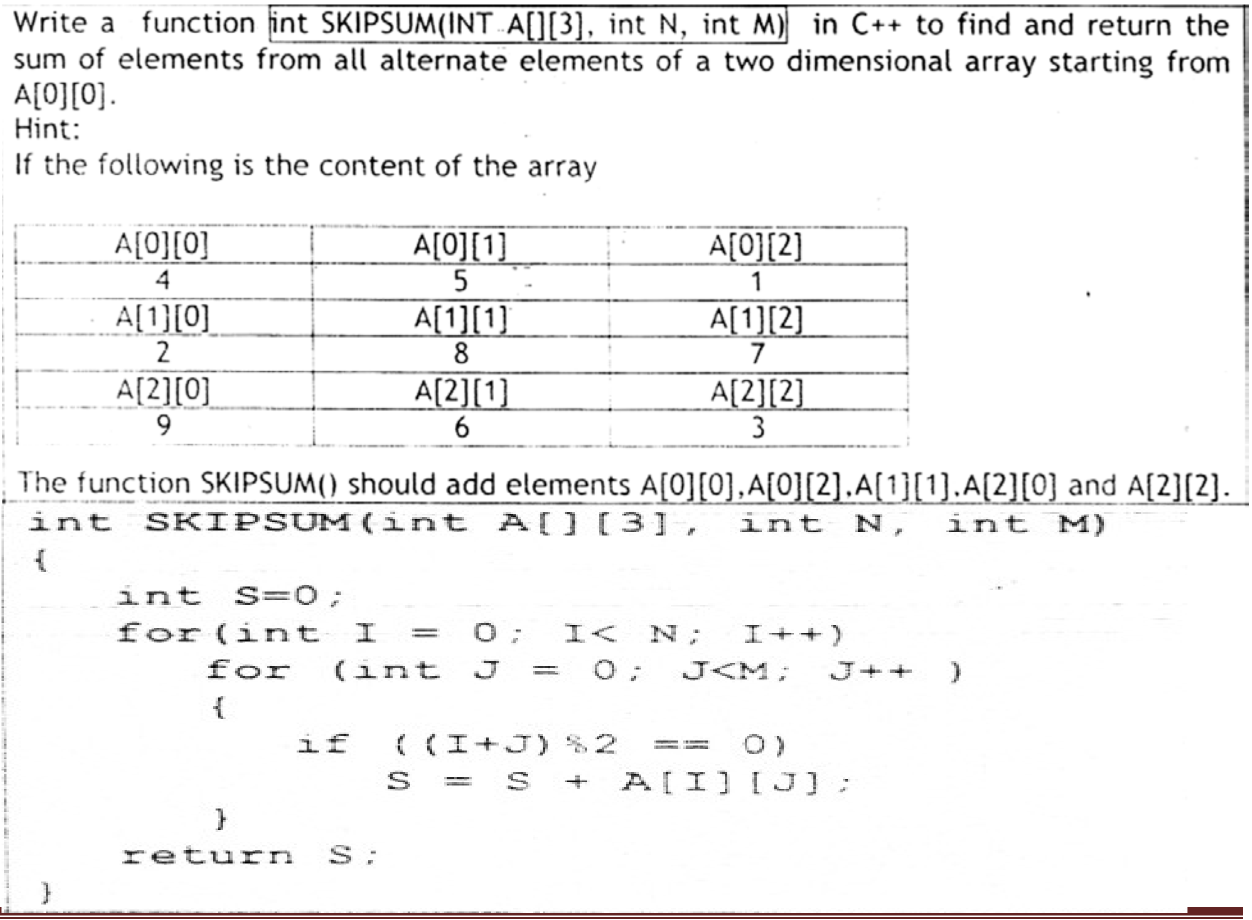
37 19 28

62 13 19
The output will be
23 76 19 62 19

Ans:
```
void ALTERNATE (int A [] [3], int N, int M)
      {
      int T=0;
      for (int I=0 ; I<N; I++)
      for (int J=0 ; J<M ; J++)
      {
      if (T%2= =0)
      cout<<A[I] [J]<<" ";
      T++ ;
      }
      }
```

Q3. Write a DSUMO function in C++ to find sum of Diagonal Elements from a NxN Matrix. (Assuming that the N is a odd number)

Ans.
```
      void DSUM (int A[] [100], int N)
      {
      int SUMR =0, SUML=0;
      for (int i=0; i<N; i++)
      {
      SUMR = SUMR + A[i] [i] ;
      SUML = SUML + A[i] [N-1-i] ;
      }
      cout<< "Sum of Right Diagonal Elements = "<<SUMR<<end1;
      cout<< "Sum of Left Diagonal Elements = "<<SUML<<end1;
      }
```



Write a function int SKIPSUM(INT A[][3], int N, int M) in C++ to find and return the sum of elements from all alternate elements of a two dimensional array starting from A[0][0].

Hint:

If the following is the content of the array

A[0][0]	A[0][1]	A[0][2]
4	5	1
A[1][0]	A[1][1]	A[1][2]
2	8	7
A[2][0]	A[2][1]	A[2][2]
9	6	3

The function SKIPSUM() should add elements A[0][0],A[0][2],A[1][1],A[2][0] and A[2][2].

```
int SKIPSUM(int A[][3], int N, int M)
{
    int S=0;
    for(int I = 0; I< N; I++)
        for (int J = 0; J<M; J++ )
        {
            if ((I+J)%2 == 0)
                S = S + A[I][J];
        }
    return S;
}
```

Q5. Define a function SWAPCOL( ) in C++ to swap (interchange) the first column elements with the last column elements, for a two dimensional integer array passed as the argument of the function.

Example: If the two dimensional array contains

2 1 4 9
1 3 7 7
5 8 6 3
7 2 1 2

After swapping of the content of 1st column and last column, it should be:

9 1 4 2
7 3 7 1
3 8 6 5
2 2 1 7

Ans:
```
void SWAPCOL(int A[ ][100], int M, int N)
{
int Temp, I;
for (I=0;I<M;I++)
{
Temp = A[I][0];
A[I][0] = A[I][N-1];
A[I][N-1] = Temp;
}
 }
```

Q6. Write a function in C++ which accepts an integer array and its size as arguments and assign the elements into a two dimensional array of integers in the following format

If the array is 1,2,3,4,5,6
The resultant 2D array is

1 2 3 4 5 6
0 1 2 3 4 5
0 0 1 2 3 4
0 0 0 1 2 3
0 0 0 0 1 2
0 0 0 0 0 1

if the array is 1,2,3
The resultant 2D array is

1 2 3
0 1 2
0 0 1

Ans: //Logic : Condition for putting the value is the position (i<=j) of 2D array otherwise put zero
```
void Change2Darray(int x[], int size)
{
      for(i=0; i<size; i++)
      {
           int k=0;
           for(int j=0; j< size; j++)
           {
                 if(i<=j)
                {
                        y[i][j]=x[k];
                        k++;
                }
                else
```

```
                            y[i][j]=0;
                    }
                }
                for(i=0; i< size; i++)
                {
                    cout<<"\n";
                    for(int j=0; j< size; j++)
                            cout<<y[i][j]<<"   ";
                 }
        }
```

Q7. Write a function in C++ which accepts an integer array of double dimensional with its size as arguments and displays the total numbers of odd, even and prime numbers in the array. Example : if the following integer array will be passed to the function, i.e.

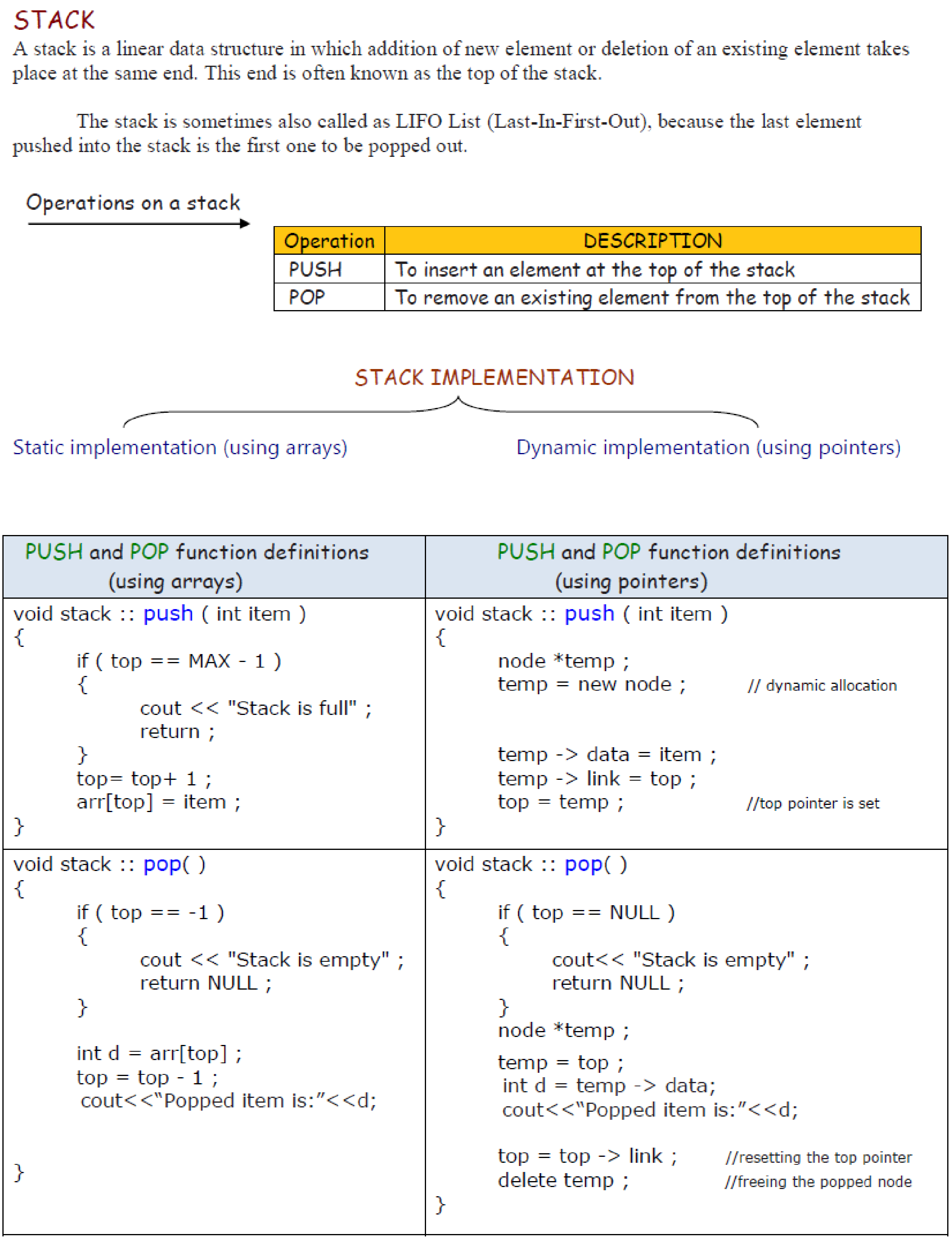
6	4	13	19	5
7	3	8	11	51
9	12	23	4	6
21	29	18	9	10
28	5	12	2	6

Then the output should be :   The total odd numbers are : 13
                                  The total odd numbers are : 12
                                  The total odd numbers are : 10

```
Ans : void numcheck( int arr[ ][ ], int m, int n)
        {
                int i, j, oddt=0, event=0, primet=0, nf, k;
                for(i=0; i<m; i++)
                {
                        for(j=0; j<n; j++)
                        {
                                if(arr[i][j] % 2 = = 0)
                                        event++;
                                else
                                        oddt++;
                                 nf=0;
                                for(k=1; k<=arr[i][j]; k++)
                                {
                                        if(arr[i][j] % k = = 0)
                                                nf++;
                                }
                                if(nf = = 2)
                                        primet++;
                        }
                }
                cout<<"\n The total odd numbers are : "<<oddt;
                cout<<"\n The total even numbers are : "<<event;
                cout<<"\n The total prime numbers are : "<<primet;
          }
```

Dynamic Allocation of Objects Long Answer Questions (4 marks)

Note : Insertion at the beginning of Linked List (Push operation in Stack), Insertion at the end of Linked List (Insertion in a Queue), Deletion from the beginning of Linked List (Pop operation in Stack as well as Deletion of node in a Queue) are in the syllabus. So, only the logic of these three functions should be practiced. The logic and the way to solve these functions are given below. Practice them.



STACK

A stack is a linear data structure in which addition of new element or deletion of an existing element takes place at the same end. This end is often known as the top of the stack.

The stack is sometimes also called as LIFO List (Last-In-First-Out), because the last element pushed into the stack is the first one to be popped out.

Operations on a stack

Operation	DESCRIPTION
PUSH	To insert an element at the top of the stack
POP	To remove an existing element from the top of the stack

STACK IMPLEMENTATION

Static implementation (using arrays)

Dynamic implementation (using pointers)

PUSH and POP function definitions (using arrays)	PUSH and POP function definitions (using pointers)

void stack :: push (int item)
{
if (top == MAX - 1)
{
cout << "Stack is full" ;
return ;
}
top= top+ 1 ;
arr[top] = item ;
}
	void stack :: push (int item)
	{
	node *temp ;
	temp = new node ; // dynamic allocation
	temp -> data = item ;
	temp -> link = top ;
	top = temp ; //top pointer is set
	}

void stack :: pop( )
{
if (top == -1)
{
cout << "Stack is empty" ;
return NULL ;
}
int d = arr[top] ;
top = top - 1 ;
cout<<"Popped item is:"<<d;
}
	void stack :: pop( )
	{
	if (top == NULL)
	{
	cout<< "Stack is empty" ;
	return NULL ;
	}
	node *temp ;
	temp = top ;
	int d = temp -> data;
	cout<<"Popped item is:"<<d;
	top = top -> link ; //resetting the top pointer
	delete temp ; //freeing the popped node
	}

QUEUE

Queue is linear data structure that permits insertion of new element at one end (called rear of the queue) and deletion of an existing element at the other end (called front of the queue). Queues are sometimes also called **FIFO List** (First-In-First-Out).

Operations on a stack →

Operation	DESCRIPTION
Insertion	To add a new element at the rear
Deletion	To remove an element from the front

QUEUE IMPLEMENTATION

Static implementation (using arrays)

Dynamic implementation (using pointers)

INSERT and DELETE function definitions (using arrays)

```
void queue :: insertq ( int item )
{
        if ( rear == MAX - 1 )
        {
                cout << "Queue is full" ;
                return ;
        }

        rear = rear + 1 ;
        arr[rear] = item ;

        if ( front == -1 )
                front = 0 ;

}
```

```
void queue :: delq( )
{
        if ( front == -1 )
        {
                cout << "Queue is Empty" ;
                return NULL ;
        }

        int d = arr[front] ;
        arr[front] = 0 ;
        if ( front == rear )
                 front = rear = -1 ;
        else
                 front = front + 1; ;

        cout<<“deleted item is:”<<d;
}
```

INSERT and DELETE function definitions (using pointers)

```
void queue :: insertq ( int item )
{
        node *temp ;
        temp = new node ;          // dynamic allocation

        temp -> data = item ;
        temp -> link = NULL ;

        if ( front == NULL )
        {
                rear = front = temp ;
                return ;
        }

        rear -> link = temp ;
        rear = rear -> link ;
}
```

```
void queue :: delq( )
{
        if ( front == NULL )
        {
                cout << "Queue is empty" ;
                return NULL ;
        }

        node *temp ;
        int d = front  -> data ;
        cout<<“deleted item is:”<<d;

        temp = front ;
        front = front -> link ;
        delete temp ;          //freeing the deleted node

}
```

1. Write a function QDELETE( ) in c++ to perform delete operation on a Linked Queue, which contains passenger no and passenger name. Consider the following definition of node in the code

```
struct node
{
long int Pno ;
char Pname[20];
node *Link ;
}
```

Ans.
```
class Queue
{
NODE *Front, *Rear;
public:
Queue ( ) {Front = NULL; Rear = NULL; }
void QueAdd ( );
void QueDel ( );
void QueDis ( );
~Queue();
} ;
void Queue: :QueDel ( )
{
if (Front!=NULL)
{
NODE *Temp=Front;
cout<<Front->Pno<< " " ;
cout<<Front->Pname<< "Deleted";
Front=Front->Link;
delete Temp;
if (Front==NULL) Rear=NULL;
}
else
cout<<"Underflow ! Queue is empty. .";
}
```

Infix & Postfix Expressions Short Answer Questions (2 marks)

Q1. Evaluate the following postfix notation of expression: 25 8 3 – / 6 * 10 +

Ans:

Operator Scanned	Stack Content
25	25
8	25, 8
3	25, 8, 3
-	25, 5
/	5
6	5, 6
*	30
10	30, 10
+	40

So, the answer is 40.

(Infix to Postfix)

Q2. Convert A + (B * C – (D / E)) * F into postfix form showing stack status after every step. Ans :

Step No.	Symbol Scanned	Stack	Expression
1	A	(	A
2	+	(+	A
3	(	(+ (	A
4	B	(+ (	A B
5	*	(+ (*	A B
6	C	(+ (*	A B C
7	–	(+ (-	A B C *
8	(	(+ (- (	A B C *
9	D	(+ (- (	A B C * D
10	/	(+ (- (/	A B C * D
11	E	(+ (- (	A B C * D E
12	)	(+ (	A B C * D E / –
13	)	(	A B C * D E / – +
14	*	(*	A B C * D E / – +
15	F	(*	A B C * D E / – + F
16	)		A B C * D E / – + F *

So, the postfix form is: A B C * D E / – + F *

Q3. Convert NOT A OR NOT B AND NOT C into postfix form. Ans :

Step No.	Symbol Scanned	Stack	Expression
1		(	
2	NOT	(NOT	
3	A	(NOT	A
4	OR	(OR	A NOT
5	NOT	(OR NOT	A NOT
6	B	(OR NOT	A NOT B
7	AND	(OR AND	A NOT B NOT
8	NOT	(OR AND NOT	A NOT B NOT
9	C	(OR AND	A NOT B NOT C NOT
10		)	A NOT B NOT C NOT AND OR

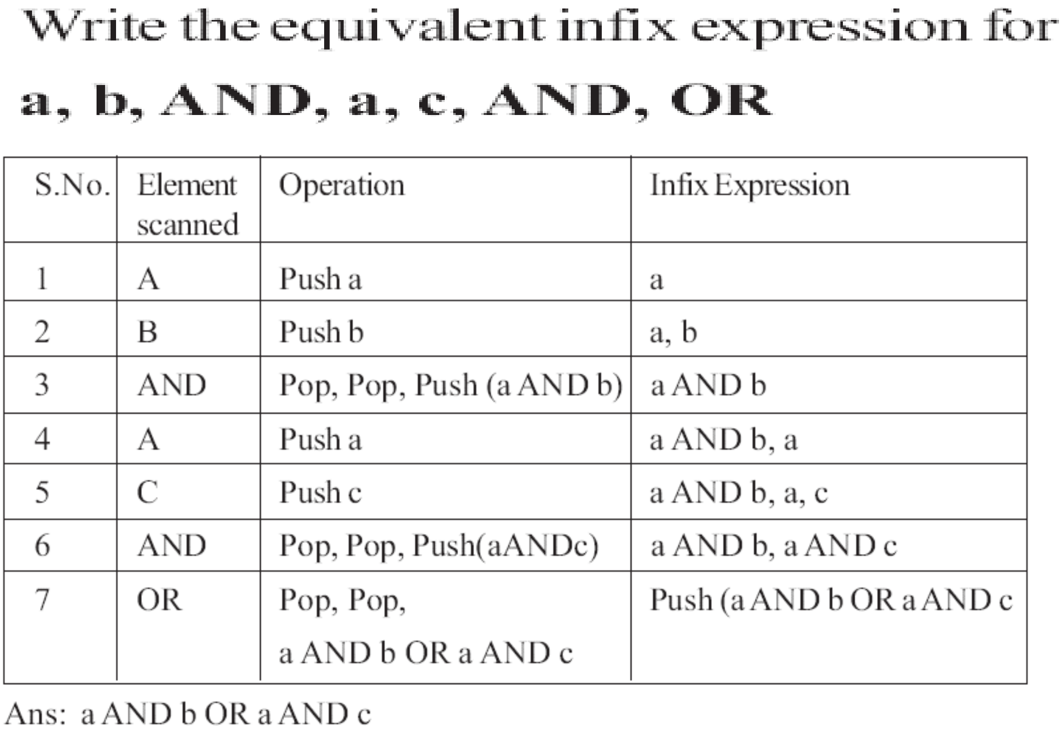
So, the postfix form is: A NOT B NOT C NOT AND OR

Q4. Evaluate the following POSTFIX notation. Show status of Stack after every step of evaluation (i.e. after each operator): True, False, NOT, AND, False, True, OR, AND

Ans.

Element Scanned	Stack Status
True	True
False	True, False
NOT	True, True
AND	True
False	True, False
True	True, False, True
OR	True, True
AND	True

Final Answer: True



Write the equivalent infix expression for

a, b, AND, a, c, AND, OR

S.No.	Element scanned	Operation	Infix Expression
1	A	Push a	a
2	B	Push b	a, b
3	AND	Pop, Pop, Push (a AND b)	a AND b
4	A	Push a	a AND b, a
5	C	Push c	a AND b, a, c
6	AND	Pop, Pop, Push(aANDc)	a AND b, a AND c
7	OR	Pop, Pop, a AND b OR a AND c	Push (a AND b OR a AND c

Ans: a AND b OR a AND c

DATA FILE HANDLING

SOME MPORTANT POINTS TO REFRESH THE CONCEPT

1.

Type of Link	Stream Class	Example Declaration
File-to-Memory	ifstream	ifstream fin;
Memory-to-File	ofstream	ofstream fout;
File-to-Memory / Memory-to-File	fstream	fstream finout;

2. Syntax of open () - <stream type>.open(<data file>, <file modes>);
   Example:- fin.open("MARK.DAT", ios::in | ios::binary);
   fout.open("MARK.DAT", ios::out | ios::app | ios::binary);
3. Different file modes are : ios::in, ios::out, ios::ate, ios::app, ios::trunc, ios::binary, ios::nocreate, ios::noreplace.
4. Syntax of read() - <stream type>.read((char *) & buffer, sizeof(buffer));
   Example : fin.read((char *) & stud, sizeof(stud)); // stud is an object/class
5. Syntax of write() - <stream type>.write((char *) & buffer, sizeof(buffer));
   Example : fout.write((char *) & stud, sizeof(stud)); // stud is an object/class
6. Detecting EOF : Example :- fin.eof()
7. Common forms of file pointer functions:
   (a) seekg() – <istream object>.seekg(long, seek_dir);
   Example : - fin.seekg(30 , ios::beg);
   (b) seekp() – <ostream object>.seekp(long, seek_dir);
   Example : - fout.seekp(–1 * sizeof(stud), ios::cur);
   (c) tellg() / tellp() returns the position in bytes of get_pointer/put_pointer.
   (d) Different definitions of seek_dir are ios::beg (refers to beginning of the file), ios::cur(refers to current position in the file), ios::end (refers to end of the file).
8. Closing a file : stream_object.close(); Example : fin.close();

Very Short Answer Questions (1 mark)

Q1. Observe the program segment given below carefully and fill the blanks marked as Statement 1 and Statement 2 using seekg()/seekp() functions for performing the required task.

```
#include <fstream.h>
class Item
{
     int Ino;char Item[20];
public:
     //Function to search and display the content of a particular record
     void Search(int );
     //Function to modify the content of a particular record number
     void Modify(int);
};
void Item::Search(int RecNo)
{
          fstream File;
          File.open("STOCK.DAT",ios::binary|ios::in);
          ______________________
          //Statement 1
          File.read((char*)this,sizeof(Item));
          cout<<Ino<<"==>"<<Item<<endl;
          File.close();
}
void Item::Modify(int RecNo)
{
          fstream File;
          File.open("STOCK.DAT",ios::binary|ios::in|ios::out);
          cout>>Ino;cin.getline(Item,20);
                    ______________________
//Statement 2
          File.write((char*)this,sizeof(Item));
          File.close();
}
```

Ans : Statement 1 – File.seekg((RecNo – 1) * sizeof(Item), ios::beg);
Statement 2 – File.seekp((RecNo – 1) * sizeof(Item), ios::beg);

Q2. Observe the program segment given below carefully and fill the blanks marked as Statement 1 and Statement 2 using seekg() and tellg() functions for performing the required task.

```
#include <fstream.h>
class Employee
{
int Eno;char Ename[20];
public:
//Function to count the total number of records
int Countrec();
};
int Item::Countrec()
```

```
    {
            fstream File;
            File.open("EMP.DAT",ios::binary|ios::in);
            ________________________
            //Statement 1

            int Bytes =_________________________                  //Statement 2

            int Count = Bytes / sizeof(Item);
            File.close();
            return Count;
    }
```

Ans: **Statement 1 – File.seekg(0,ios::end);**
Statement 2 – File.tellg();

Q3. A file named as “STUDENT.DAT” contains the student records, i.e. objects of class student. Assuming that the file is just opened through the object FILE of fstream class, in the required file mode, write the command to position the put pointer to point to the beginning of the second record from the last record.

Ans : FILE.seekp(– 2 * sizeof(student), ios::end);

Q4. A file named as “EMPLOYEE.DAT” contains the student records, i.e. objects of class employee. Assuming that the file is just opened through the object FILE of fstream class, in the required File mode, write the command to position the get pointer to point to eighth record from the beginning.

Ans : File.seekg(8,ios::beg);

Short Answer Questions (2 marks)

Q1. Write a function in C++ to count the number of alphabets present in a text file “DATA.TXT”.

Ans:
```
void count( )
{
        ifstream file("DATA.TXT"); char
        ch; int total=0; while(file.get(ch))
        {
                if(isalpha(ch))
                        total++;
        }
        cout<<"\n The Total no of alphabets are"<<total; file.close( );
}
```

Q2. Write a function in C++ to count the number of vowels present in a text file “STORY.TXT”.

Ans:
```
void vowelcount( )
{
        ifstream file("STORY.TXT");
        int n=0;
        while(file.get(ch))
        {
                if(ch=='a'|| ch=='A'|| ch=='e'|| ch=='E'|| ch=='i'|| ch=='I'|| ch=='o'||
                ch=='O'|| ch=='u'||ch=='U')
                        n++ ;
        }
        cout<<"\n Total no. of vowels are"<<n; file.close(); }
```

Q3. Write a function in C++ to count the number of words in a text file NOTES.TXT.

Ans:

```
void wordcount( )
{
          ifstream f1(“NOTES.TXT”); char
          ch;
          int tot=0; f1.seekg(0);
          while(f1.get(ch))
          {
                    if( ch==’ ‘|| ch==’.’|| ch==’?’)
                              tot++;
          }
          cout<<”\n total no. of words are “<<tot;
          f1.close();
}
```

Q4. Write a function in C++ to count the number of lines present in a text file “STORY.TXT”.

Ans :

```
void CountLine()
{
          ifstream FIL(“STORY.TXT”);
          int LINES=0;
          char STR[80];
          while (FIL.getline(STR,80))
                    LINES++;
          cout<<”No. of Lines:”<<LINES<<endl;
          FIL.close();
}
```

Q5. Write a function CountYouMe( ) in c++ which reads the contents of a text file story.txt and counts the words You and Me (not case senitive)

For example, if the file contains

You are my best friend.

You and me make a good team.

The function should display the output as

Count for you: 2

Count for me: 1

Ans:

```
void COUNTYouMe ( )
{
ifstream Fil ("story.txt") ;
char STR [10] ;
int countyou = 0 ;
int countme = 0;
while (!Fil.eof ( ))
{
Fil>>STR;
if (strcmpi (STR, "Me") = =0 )
countme++ ;

if(strcmpi (STR, "You")  = =0)
countyou++;
}
cout<<"Count for You : "<<countyou<<end1;
cout<<"Count for Me : "<<countme<<end1;
Fil.close( ) ; //Ignore
}
```

Q6. Write a function in C++ to read the content of a text file "DELHI.TXT' and display all those lines on screen, which are either starting with 'D' or starting with 'M'.

Ans.
```
void DispDorM()
{
if stream File("DELHI.TXT");
char Str[80];
while(File.getline(Str,80))
{
if(Str[0]='D' || Str[0]='M')
cout«Str«endl;
}
File.close(); //Ignore
}
```

Short Answer Questions (3 marks)

Q1. Write a function in C++ to read & display the details of all the members whose membership type is ‘L’ or ‘M’ from a binary file “CLUB.DAT”. Assume the binary file contains object of class CLUB.

```
class CLUB
{
    int mno;
    char mname[20];
    char type;
public:
    void register ( );
    void dis();
    char whattype( )
    {    return type;}
};
```

Ans :
```
void show( )
{
    ifstream file;
    CLUB c1;
    char s;
    file.open(“CLUB.DAT”, ios::in | ios::binary);
    while( file.read((char*)&c1, sizeof (c1)))
    {
        s = c1.whattype( );
        if((s==’L’) || (s==’M’))
                c1.dis( );
    }
    File.close();
}
```

Q2. Write a function TRANSFER( ) in C++, that would copy all records which are having area as ”DEL” from PHONE.DAT to PHONBACK.DAT.

Ans:

```
void transfer( )
{   ifstream Fin;
    Ofstream Fout;
    Phonlist ph;
    Fin.open(“PHONE.DAT”, ios::in | ios::binary);
    Fout.open(“PHONBACK.DAT”, ios::out | ios:: binary);
    while(Fin.read((char*)&ph, sizeof(ph)))
    {   if(ph.check(“DEL”) == 0)
                Fout.write((char*)&ph, sizeof(ph));
    }
    Fin.close();
    Fout.close();
}
```

Q3. Write a function in C++ to modify the name of a student whose roll number is entered during the execution of the program. Search the accepted roll number from a binary file “STUDENT.DAT” and modify the name, assuming the binary file is containing the objects of the following class.

```
class STUD
{
        int Rno;
        char Name[20];
public:
        int srno(){ return Rno;}
        void Enter(){gets(Name);}
        void Display(){cout<<Rno<<Name<<endl;}
        };
```

Ans :

```
void   Modifyname()
{        fstream   File;
         File.open(“STUDENT.DAT”, ios::binary | ios::in | ios::out);
         STUD s;
         int mrno;
         cout<<”\nEnter the roll number to modify : “;
         cin>>mrno;
         while (File.read((char*)&s, sizeof(s)))
         {
                 if(s.srno() = = mrno)
                 {
                         cout<<”\n Modify the name
                         “; s.Enter();
                         File.seekp( –1 * sizeof(s),
                         ios::cur); File.write((char
                         *)&s, sizeof(s));
                         }File.close() ; }
```

Databases and SQL **Short Answer Questions**

Q1. What is a database system? What are the advantages provided by a database system?

Ans : A database is a collection of interrelated data and a database system is basically a computer based recordkeeping system. The advantages provided by a database system are (i) reduced data redundancy (ii) Controlled data inconsistency (iii) Shared data (iv) Standardized data

(i) Secured data (vi) Integrated data.

Q2. What are the various levels of database abstraction in a database system?

Ans : The various levels of database abstraction in a database system are (i) Internal (Physical) Level (ii) Conceptual Level (iii) External (View) Level.

Q3. What is Data Independence? What are the levels of Data Independence?

Ans: Data Independence is the ability to modify a scheme definition in one level without affecting a scheme definition in the next higher level. The two levels of Data Independence are (i) Physical Data Independence (ii) Logical Data Independence.

Q4. What is a view? How is it useful?

Ans : A view is a virtual table that does not really exist in its own right but is instead derived from one or more underlying base table(s). It provides an excellent way to give people access to some but not all of the information in a table.

Q5. What is relation? Define the relational data model?

Ans : A relation is a table having atomic values, unique rows and unordered rows and columns. The relational data model represents data and relationships among data by a collection of tables known as relation, each of which has a number of columns with unique names.

Q6. Define : (i) Tuple (ii) Attribute (iii) Degree (iv) Cardinality. Ans : (i) Tuple – The rows of tables (relations) are referred as tuples.

(ii) Attribute – The columns of tables (relations) are referred as attributes.

(iii) Degree – The number of attributes in a relation(table) is called degree of a relation.

(iv) Cardinality – The number of tuples in a relation(table) is called cardinality of a relation. Q7. What do you understand by Primary Key & Candidate Keys?

Ans : An attribute or set attributes which are used to identify a tuple uniquely is known as Primary Key. If a table has more than one such attributes which identify a tuple uniquely than all such attributes are known as Candidate Keys.

Q8. Define alternate key. What do you understand by foreign key?

Ans : A candidate key that is not the primary key is called an alternate key. A non-key attribute, whose values are derived from the primary key of some other table, is known as foreign key in its current table.

Q9. What is Data Definition Language? Give examples of some DDL commands.

Ans : The SQL Data Definition Language provides commands for defining relation schemas, deleting relations, creating indexes and modifying relation schemas. The examples of some DDL commands are CREATE TABLE, ALTER TABLE, DROP TABLE, DROP VIEW commands etc.

Q10. What is Data Manipulation Language? Give examples of some DML commands.

Ans : The SQL Data Manipulation Language includes a query language to insert, delete and modify tuples in the database. The examples of some DML commands are SELECT, INSERT, DELETE, UPDATE commands etc.

Q11. What is SQL? Write the various processing capabilities of SQL.

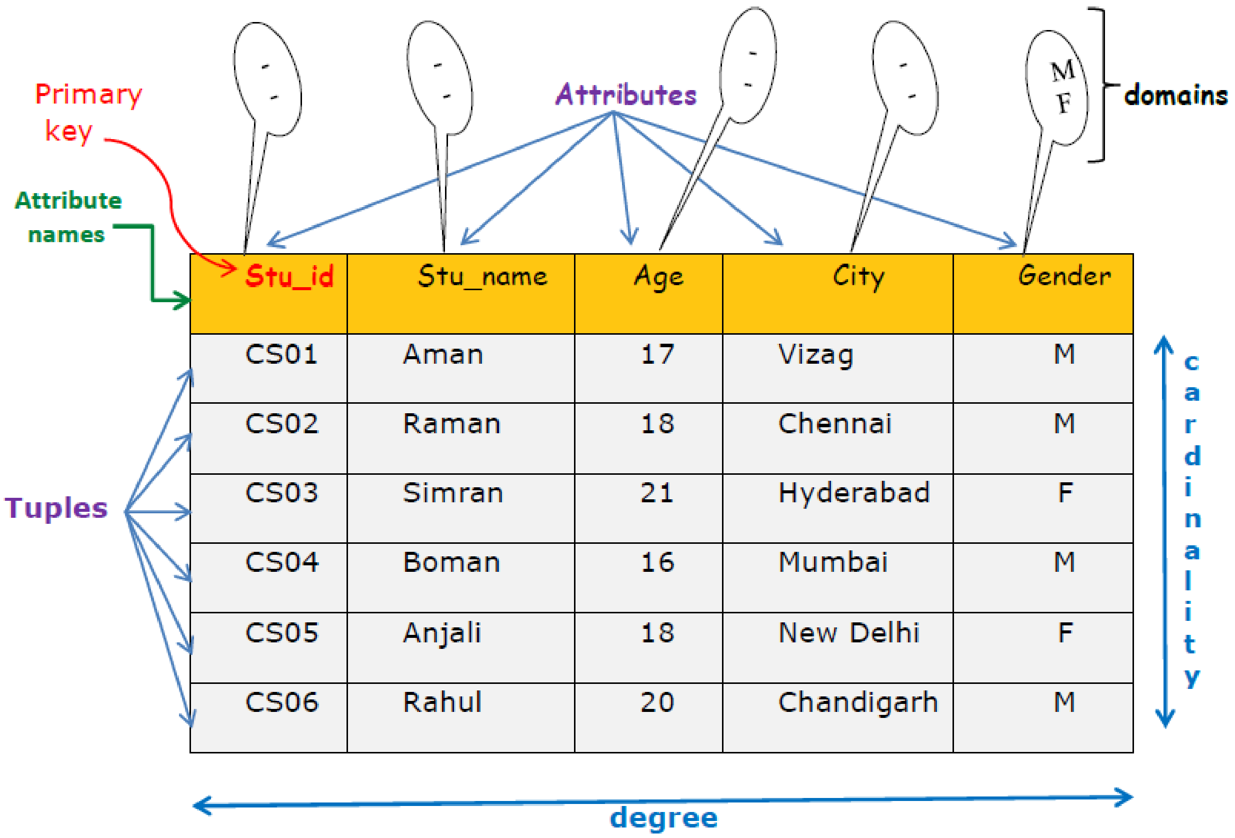
Ans : SQL is a language that enables you to create and operate on relational databases. The various processing capabilities of SQL are data definition language, data manipulation language, view definition, authorization, integrity and transaction control.

Relational Model Terminology

The following is the list of relational terms and their corresponding meanings:

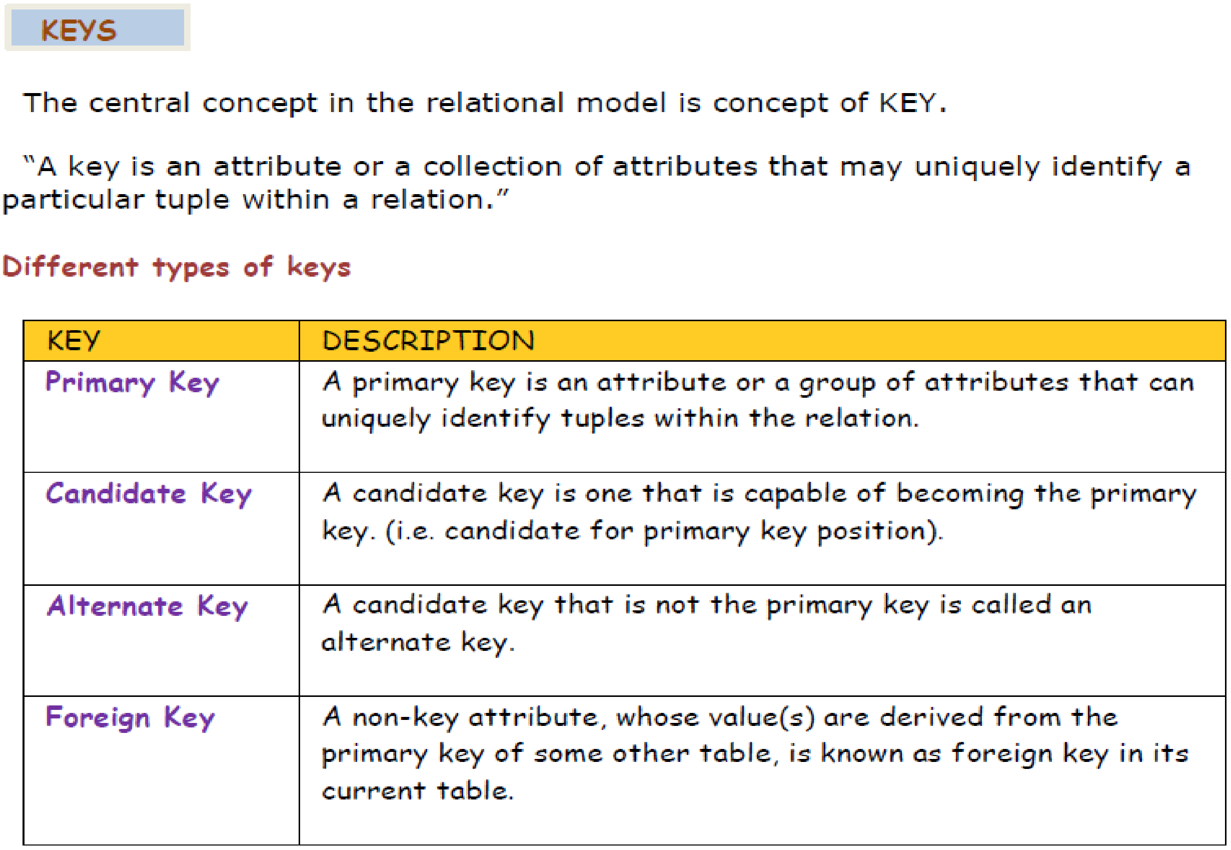
Formal Relational Term	Their Meanings
Relation	Table
Tuple	Row (or record)
Attribute	Column (or field)
Cardinality	No. of rows
Degree or Arity	No. of columns
Domain	Set of legal values
Primary key	Unique identifier
Foreign Key	Identifier used to reference another table.

To understand these terms, let us consider an instance of a **STU** relation.



Stu_id	Stu_name	Age	City	Gender
CS01	Aman	17	Vizag	M
CS02	Raman	18	Chennai	M
CS03	Simran	21	Hyderabad	F
CS04	Boman	16	Mumbai	M
CS05	Anjali	18	New Delhi	F
CS06	Rahul	20	Chandigarh	M

- In the above STU Relation, there are 6 tuples (i.e. cardinality = 6) and 5 attributes (i.e. degree = 5).
- Stu_id, Stu_name, Age, City, and Gender are the attribute names.
- The first tuple contains (CS01, 'Aman' ,17, 'vizag' , 'M') as its values.
- The domain of the of the gender attribute is (M,F).



KEYS

The central concept in the relational model is concept of KEY.

"A key is an attribute or a collection of attributes that may uniquely identify a particular tuple within a relation."

Different types of keys

KEY	DESCRIPTION
Primary Key	A primary key is an attribute or a group of attributes that can uniquely identify tuples within the relation.
Candidate Key	A candidate key is one that is capable of becoming the primary key. (i.e. candidate for primary key position).
Alternate Key	A candidate key that is not the primary key is called an alternate key.
Foreign Key	A non-key attribute, whose value(s) are derived from the primary key of some other table, is known as foreign key in its current table.

SYNTAX of some SQL COMMANDS :

1. SELECT column list FROM <table name> WHERE <condition> GROUP BY <column name(s)> HAVING <search condition> ORDER BY column name;
2. INSERT INTO <table name> [<column list>] VALUES (<value>, <value>, ...);
3. DELETE FROM <table name> [WHERE <predicate>];
4. UPDATE <table name> SET <column name> = <new value> [WHERE <predicate>];
5. CREATE TABLE <table name> (<column name> <data type> [(size)] <column constraint>, <column name> <data type> [(size)] <column constraint>, ... <table constraint> (<column name> [<column name> ...]) ...);
6. CREATE VIEW <view name> [(<column name>, <column name>,...)] AS <SELECT command>;
7. ALTER TABLE <table name> ADD / MODIFY <column name> <data type> <size>;
8. DROP TABLE <table name>;
9. DROP VIEW <view name>;

Long Answer Questions

Q1. Consider the following tables ACTIVITY and COACH. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table: ACTIVITY

ACode	ActivityName	ParticipantsNum	PrizeMoney	ScheduleDate
1001	Relay 100x4	16	10000	23-Jan-2004
1002	High jump	10	12000	12-Dec-2003
1003	Shot Put	12	8000	14-Feb-2004
1005	Long Jump	12	9000	01-Jan-2004
1008	Discuss Throw	10	15000	19-Mar-2004

Table: COACH

PCode	Name	ACode
1	Ahmad Hussain	1001
2	Ravinder	1008
3	Janila	1001
4	Naaz	1003

(i) To display the name of all activities with their Acodes in descending order.

(ii) To display sum of PrizeMoney for each of the Number of participants groupings (as shown in column ParticipantsNum 10,12,16).

(iii) To display the coach's name and ACodes in ascending order of ACode from the table COACH.

(iv) To display the content of the GAMES table whose ScheduleDate earlier than 01/01/2004 in ascending order of ParticipantNum.

(v) SELECT COUNT(DISTINCT ParticipantsNum) FROM ACTIVITY;

(vi) SELECT MAX(ScheduleDate),MIN(ScheduleDate) FROM ACTIVITY;

(vii) SELECT SUM(PrizeMoney) FROM ACTIVITY;

(viii) SELECT DISTINCT ParticipantNum FROM COACH;

Ans : (i) SELECT ActivityName, ACode FROM ACTIVITY ORDER BY Acode DESC;

(ii) SELECT SUM(PrizeMoney),ParticipantsNum FROM ACTIVITY GROUP BY ParticipantsNum;

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

(iv) SELECT * FROM ACTIVITY WHERE ScheduleDate<'01-Jan-2004' ORDER BY ParticipantsNum;

(v) 3

(vi) 19-Mar-2004 12-Dec-2003

(vii) 54000

(viii) 1
6

1
0

1
2

Q2. Consider the following tables GAMES and PLAYER. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table: GAMES

GCode	GameName	Number	PrizeMoney	ScheduleDate
101	Carom Board	2	5000	23-Jan-2004
102	Badminton	2	12000	12-Dec-2003
103	Table Tennis	4	8000	14-Feb-2004
105	Chess	2	9000	01-Jan-2004
108	Lawn Tennis	4	25000	19-Mar-2004

Table: PLAYER

PCode	Name	Gcode
1	Nabi Ahmad	101
2	Ravi Sahai	108
3	Jatin	101
4	Nazneen	103

(i) To display the name of all Games with their Gcodes.

(ii) To display details of those games which are having PrizeMoney more than 7000. (iii)To display the content of the GAMES table in ascending order of ScheduleDate.

(iv) To display sum of PrizeMoney for each of the Number of participation groupings (as shown in column Number 2 or 4).

(v) SELECT COUNT(DISTINCT Number) FROM GAMES;

(vi) SELECT MAX(ScheduleDate),MIN(ScheduleDate) FROM GAMES;

(vii) SELECT SUM(PrizeMoney) FROM GAMES;

(viii) SELECT DISTINCT Gcode FROM PLAYER;

Ans : (i) SELECT GameName,Gcode FROM GAMES;

(ii) SELECT * FROM GAMES WHERE PrizeMoney>7000;

(iii) SELECT * FROM GAMES ORDER BY ScheduleDate;

(iv) SELECT SUM(PrizeMoney),Number FROM GAMES GROUP BY Number;

(v) 2

(vi) 19-Mar-2004 12-Dec-2003

(vii) 59000

(viii)
```
1
0
1
1
0
3
1
0
8
```

Q3. Consider the following tables HOSPITAL. Give outputs for SQL queries (i) to (iv) and write SQL commands for the statements (v) to (viii).

No	Name	Age	Department	Dateofadmin	Charge	Sex
1	Arpit	62	Surgery	21/01/06	300	M
2	Zayana	18	ENT	12/12/05	250	F
3	Kareem	68	Orthopedic	19/02/06	450	M
4	Abhilash	26	Surgery	24/11/06	300	M
5	Dhanya	24	ENT	20/10/06	350	F
6	Siju	23	Cardiology	10/10/06	800	M
7	Ankita	16	ENT	13/04/06	100	F
8	Divya	20	Cardiology	10/11/06	500	F
9	Nidhin	25	Orthopedic	12/05/06	700	M
10	Hari	28	Surgery	19/03/06	450	M

(i) Select SUM(Charge) from HOSPITAL where Sex=’F’;

(ii) Select COUNT(DISTINCT Department) from HOSPITAL;

(iii) Select SUM(Charge) from HOSPITAL group by Department;

(iv) Select Name from HOSPITAL where Sex=’F’ AND Age > 20;

(v) To show all information about the patients whose names are having four characters only.

(vi) To reduce Rs 200 from the charge of female patients who are in Cardiology department.

(vii) To insert a new row in the above table with the following data : 11, ‘Rakesh’, 45, ‘ENT’, {08/08/08}, 1200, ‘M’

(viii) To remove the rows from the above table where age of the patient > 60.

Ans : (i) 1200

(ii) 4

(iii) 1050
700
11
50
13
00

(iv) Dhanya

(v) SELECT * FROM HOSPITAL WHERE NAME LIKE “_ _ _ _”;

(vi) UPDATE HOSPITAL SET CHARGE = CHARGE – 200 WHERE (DEPARTMENT = ‘CARDIOLOGY’ AND SEX = ‘F’);

(vii) INSERT INTO HOSPITAL VALUES(11,‘Rakesh’,45,‘ENT’,{08/08/08}, 1200, ‘M’);

(viii) DELETE FROM HOSPITAL WHERE AGE > 60;

Q4. Consider the following tables BOOKS. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table : BOOKS

B_Id	Book_Name	Author_Name	Publisher	Price	Type	Quantity
C01	Fast Cook	Lata Kapoor	EPB	355	Cookery	5
F01	The Tears	William Hopkins	First	650	Fiction	20
T01	My C++	Brain & Brooke	FPB	350	Text	10
T02	C++ Brain	A.W.Rossaine	TDH	350	Text	15
F02	Thuderbolts	Anna Roberts	First	750	Fiction	50

i). To list the names from books of Text type.
ii). To display the names and price from books in ascending order of their
price. iii). To increase the price of all books of EPB publishers by 50.
iv). To display the Book_Name, Quantity and Price for all C++
books. v). Select max(price) from books;
vi). Select count(DISTINCT Publishers) from books where Price >=400;
vii). Select Book_Name, Author_Name from books where Publishers =
‘First’; viii). Select min(Price) from books where type = ‘Text’;

Ans : (i) SELECT Book_Name FROM BOOKS WHERE Type = ‘Text’;
(ii) SELECT Book_Name, Price FROM BOOKS ORDER BY Price;
(iii) UPDATE BOOKS SET Price = Price + 50 WHERE Publisher = ‘EPB’;
(iv) SELECT Book_Name, Quantity, Price FROM BOOKS WHERE Book_Name LIKE ‘%C++%’;
(v) 750
(vi) 2
(vii) The Tears William
Hopkins Thuderbolts Anna
Roberts
(viii) 350

Q5. Consider the tables ITEMS & COMPANY. Write SQL commands for the statements (i) to (iv) and give the outputs for SQL queries (v) to (viii).

Table : ITEMS

ID	**PNAME**	**PRICE**	**MDATE**	**QTY**
T001	Soap	12.00	11/03/2007	200
T002	Paste	39.50	23/12/2006	55
T003	Deodorant	125.00	12/06/2007	46
T004	Hair Oil	28.75	25/09/2007	325
T005	Cold Cream	66.00	09/10/2007	144
T006	Tooth Brush	25.00	17/02/2006	455

Table : COMPANY

ID	**COMP**	**City**
T001	HLL	Mumbai
T008	Colgate	Delhi
T003	HLL	Mumbai
T004	Paras	Haryana
T009	Ponds	Noida
T006	Wipro	Ahmedabad

i). To display PNAME, PRICE * QTY only for the city Mumbai.

ii). To display product name, company name & price for those items which IDs are equal to the IDs of company.

iii). To delete the items produced before 2007.

iv). To increase the quantity by 20 for soap and paste.

v). SELECT COUNT(*) FROM ITEMS WHERE ITEMS.ID=COMPANY.ID;

vi). SELECT PNAME FROM ITEMS WHERE PRICE=SELECT MIN(PRICE) FROM ITEMS;

vii). SELECT COUNT(*) FROM COMPANY WHERE COMP LIKE “P_ _ _ _”;

viii). SELECT PNAME FROM ITEMS WHERE QTY<100;

Ans : (i) SELECT PNAME, QTY*PRICE FROM ITEMS WHERE ITEMS.ID = COMPANY.ID AND COMPANY.City=’Mumbai’;

(ii) SELECT PNAME, COMP, PRICE FROM ITEMS, COMPANY WHERE ITEMS.ID = COMPANY.ID

(iii) DELETE FROM ITEMS WHERE MDATE < {01/01/2007};

(iv) UPDATE ITEMS SET QTY = QTY + 20 WHERE PNAME = ‘Soap’ OR PNAME = ‘Paste’;

(v) 4

(vi) Soap

(vii) 2

(viii) Paste
Deodorant

Boolean Algebra

Introduction to Boolean Algebra & Laws

Boolean Algebra Rules

1 2	0 + X = X 0 . X = 0	Properties of 0
3 4	1 + X = 1 1 . X = X	Properties of 1
5 6	X + X = X X . X = X	Idempotence Law
7	(X′)′ = X	Involution Law
8 9	X + X′ = 1 X . X′ = 0	Complementarity Law
10 11	X + Y = Y + X X .Y = Y. X	Commutative Law
12 13	X + (Y + Z) = (X + Y) + Z X (YZ) = (XY) Z	Associative Law

14 15	X (Y + Z) = XY + XZ X + YZ = (X + Y) (X + Z)	Distributive Law
16 17	X + XY = X X . (X + Y) = X	Absorption Law
18	X + X′ Y = X +Y	Third Distributive Law
19 20	(X + Y) ′ = X′ . Y′ (X.Y) ′ = X′ + Y′	DeMorgan’s Theorem

Short Answer Questions (2 marks)

Q1. State and verify Demorgan's Laws algebraically.

Ans : Demorgan's Laws are : (i) (X+Y)' = X'.Y' (i) (X.Y)' = X' + Y'

Verification

(X+Y)'.(X+Y) = X'.Y'.(X+Y)
0 = X'.Y'.X + X'.Y'.Y
0 = X'.X .Y'+ X'.0
0 = 0 .Y'+ 0
0 = 0 + 0
0 = 0
L.H.S = R.H.S

(X.Y)' + (X.Y) = (X' + Y') + (X.Y)
1 = (X' + Y' + X). (X' + Y' + Y)
1 = (X' + X + Y') . (X' + 1)
1 = (1 + Y') . 1
1 = 1 . 1
1 = 1
L.H.S = R.H.S

Q2. State and algebraically verify Absorption Laws.

Ans : Absorption Laws are : (i) X+X.Y = X (ii) X . (X + Y) = X

Verification:

L.H.S = X + X . Y
= X.1 + X.Y
= X.(1 + Y)
= X.1
=X
= R.H.S

L.H.S = X . (X + Y)
= X.X + X.Y = X + X.Y
= X.(1 + Y)
= X.1
=X
= R.H.S

Q3. State Distributive Laws. Write the dual of : X + X′ Y = X + Y

Ans : Distributive laws are : (i) X . (Y + Z) = X.Y + X.Z (ii) X + Y.Z = (X + Y).(X + Z)

The dual form of X + X′ Y = X + Y is X . (X′ + Y) = X . Y

Q4. State any one form of Associative law and verify it using truth table.

Ans : Associative law states that : (i) X + (Y + Z) = (X + Y) + Z (ii) X.(Y.Z) = (X.Y).Z

Verification of X + (Y + Z) = (X + Y) + Z using truth table

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

Q5. State the principle of duality in boolean algebra and give the dual of the following:

X.Y′.Z+X.Y′.Z′+X′.Y′.Z

Ans : Principle of duality states that from every boolean relation, another boolean relation can be derived by : (i) changing each OR (+) sign to an AND (.) sign
(ii) changing each AND (.) sign to an OR (+) sign
(iii) replacing each 1 by 0 and each 0 by 1.

The dual form ofX.Y′.Z+X.Y′.Z′+X′.Y′.Z is (X+Y′+Z) . (X+Y′+Z′) . (X′+Y′+Z)

Q6. Explain about tautology and fallacy.

Ans : If result of any logical statement or expression is always TRUE or 1, it is called tautology.
If result of any logical statement or expression is always FALSE or 0, it is called fallacy.
For example, $1 + X = 1$ is a tautology and $0 . X = 0$ is a fallacy.

SOP & POS

Very Short Answer Questions (1 Mark)

Q1. Write the POS form of a Boolean function F, which is represented in a truth table as follows:

U	V	W	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Note: In POS, consider only the cases those containing 0 in F column. For the same rows, write sum of the logical variables (Ex. U i.e. 0 in U column, U' i.e. 1 for U column). Then write the product form of all such sum of logical terms.

Ans : $F(U,V,W) = (U + V + W').(U + V' +W').(U' + V + W')$

Q2. Write the SOP form of a Boolean function G, which is represented in a truth table as follows:

X	Y	Z	G
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Note: In SOP, consider only the cases those containing 1 in G column. For the same rows, write product of the logical variables (Ex. X' i.e. 0 in X column, X i.e. 1 for X column). Then write the sum form of all such product of logical terms.

Ans : $G(X,Y,Z) = (X' . Y' . Z) + (X' . Y .Z') + (X . Y' . Z) + (X . Y . Z')$

Q3. Convert the following function into canonical SOP form.

$F(A , B, C, D) = \Sigma (1, 4, 6, 8, 11, 13)$

Ans : $F(A , B, C, D) = A' B' C' D + A' B C' D' + A' B C D' + A B' C' D' + A B' C D + A B C' D$

(Note : Here $1 = 0001 \Rightarrow A' B' C' D$, $4 = 0100 \Rightarrow A' B C' D'$, $13 = 1101 \Rightarrow A B C' D$ ...)

Q4. Convert the following function into canonical SOP form.

$F(A , B, C, D) = \pi (0, 3, 7, 9, 10, 14)$

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

(Note : Here $0 = 0000 \Rightarrow A+ B+C+D$, $3 = 0011 \Rightarrow A+B+C'+D'$, $10 = 1010 \Rightarrow A'+B+C'+D$...)

Q5. Convert the following function to its equivalent SOP shorthand notation.

$F(A , B, C, D) = \pi (0, 3, 7, 9, 10, 14)$

Ans : The equivalent SOP shorthand notation of $F(A , B, C, D) = \pi (0, 3, 7, 9, 10, 14)$
is $F(A , B, C, D) = \Sigma (1, 2, 4, 5, 6, 8, 11, 12, 13, 15)$

Q6. Convert the following function to its equivalent POS shorthand notation.

$F(A , B, C, D) = \Sigma (1, 4, 6, 8, 11, 13)$

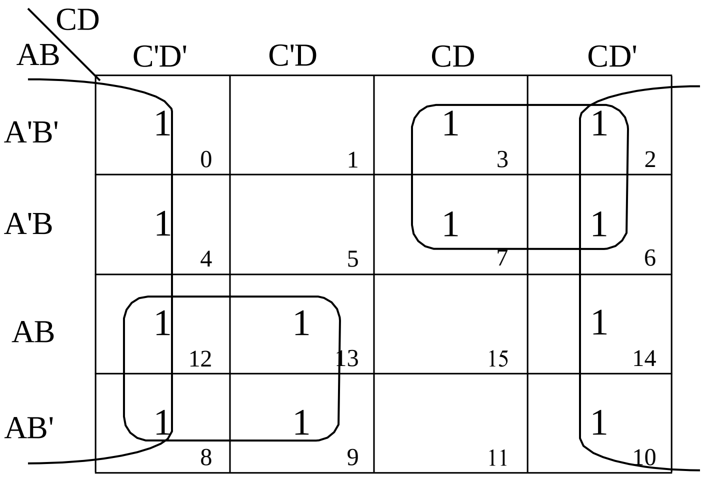
Ans : The equivalent POS shorthand notation of $F(A , B, C, D) = \Sigma (1, 4, 6, 8, 11, 13)$
is $F(A , B, C, D) = \pi (0, 2, 3, 5, 7, 9, 10, 12, 14, 15)$

Karnaugh Map Short Answer Questions (3 marks)

Q1. Reduce the following Boolean Expression using K-Map:

F(A, B, C, D) = Σ (0, 2, 3, 4, 6, 7, 8, 9, 10, 12, 13, 14)

Ans :

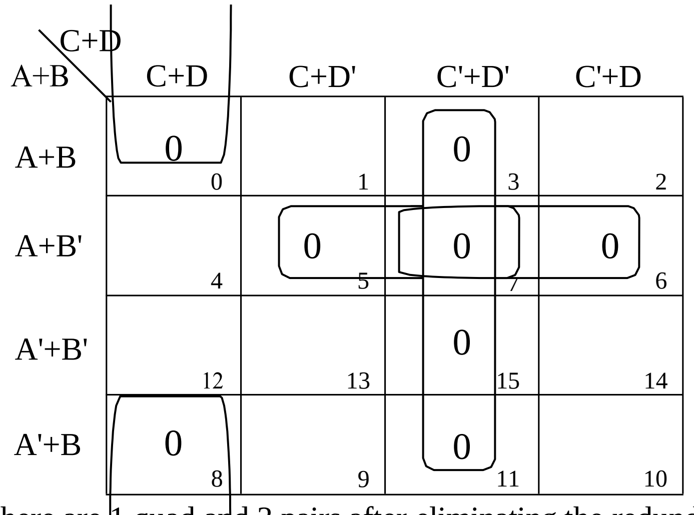


AB \ CD	C'D'	C'D	CD	CD'
A'B'	1 0	1	1 3	1 2
A'B	1 4	5	1 7	1 6
AB	1 12	1 13	15	1 14
AB'	1 8	1 9	11	1 10

There are 1 octet, 2 quads after eliminating the redundant groups. Octet (m_0, m_2, m_4, m_6, m_8, m_{10}, m_{12}, m_{14}) reduces to D' Quad (m_2, m_3, m_6, m_7)reduces to A'C Quad (m_8, m_9, m_{12}, m_{13}) reduces to AC ' Hence, F(A, B, C, D) = D' + A'C + AC '

Q2. Reduce the following Boolean Expression using K-Map: F(A, B, C, D) = π (0, 3, 5, 6, 7, 8, 11, 15)

Ans :



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

There are 1 quad and 3 pairs after eliminating the redundant groups.

Quad (M_3, M_7, M_{11}, M_{15}) reduces to C' + D'

Pair (M_5, M_7) reduces to A + B ' + D'

Pair (M_6, M_7) reduces to A + B ' + C'

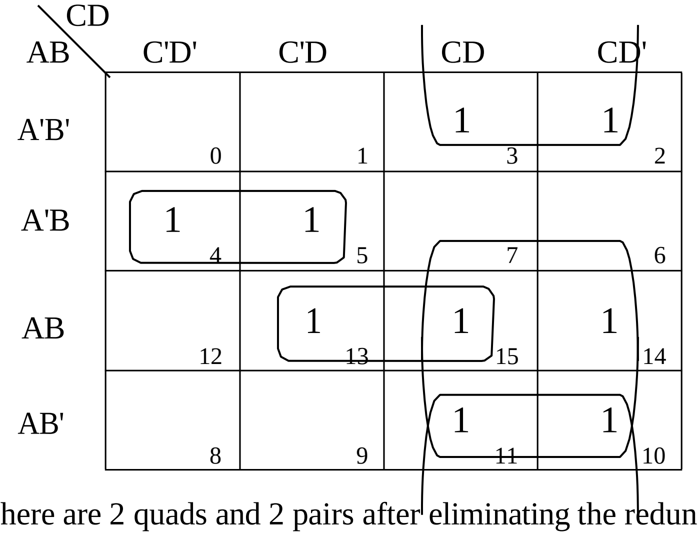
Pair (M_0, M_8) reduces to B + C + D

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

Q3. Reduce the following Boolean Expression using K-Map:

F(A, B, C, D) = Σ (2, 3, 4, 5, 10, 11, 13, 14, 15)

Ans :



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

There are 2 quads and 2 pairs after eliminating the redundant groups.

Quad 1 (m_2, m_3, m_{10}, m_{11}) reduces to B'C

Quad 2 (m_{10}, m_{11}, m_{14}, m_{15}) reduces to AC

Pair 1 (m_4, m_5) reduces to A' BC '

Pair 1 (m_{13}, m_{15}) reduces to ABD

Hence, F(A, B, C, D) = B'C + AC + A'BC ' + ABD

Q4. Reduce the following Boolean Expression using K-Map: F(A,B,C,D)=Σ(0,1,4,5,7,10,13,15)

Ans:

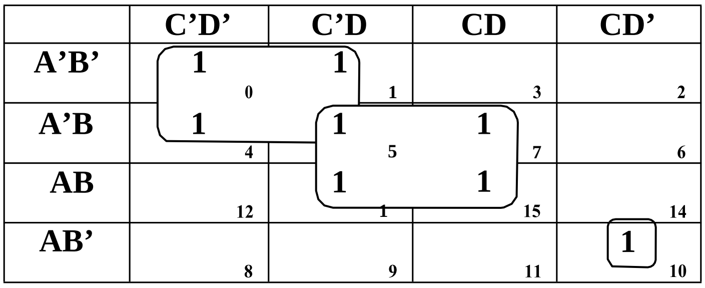
There are 2 quads and one block after eliminating the redundant groups.

Quad 1 (m_0, m_1, m_4, m_5) reduces to A' C '

Quad 2 (m_0, m_2, m_4, m_6) reduces to B D

m_{10} = A B ' C D '

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



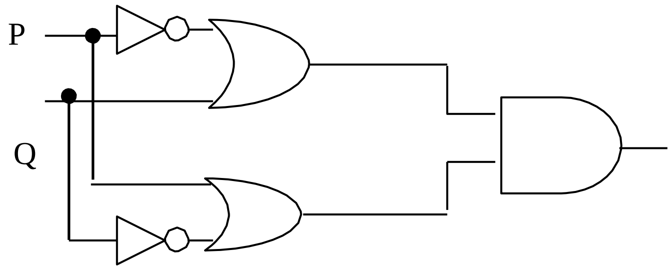
	C’D’	C’D	CD	CD’
A’B’	1 0	1 1	3	2
A’B	1 4	1 5	1 7	6
AB	12	1 1	1 15	14
AB’	8	9	11	1 10

Basic Logic Gates

Note : A gate is simply an electronic circuit which operates on one or more signals to produce an output signal. The three basic logic gates are NOT gate, OR gate, AND gate. NOR gate, NAND gate, XOR gate, XNOR gate can be derived from these three basic gates. NOR and NAND gates are called universal gates because any combinational circuit is possible using NOR or NAND gates. Practice different questions based on logic gates.

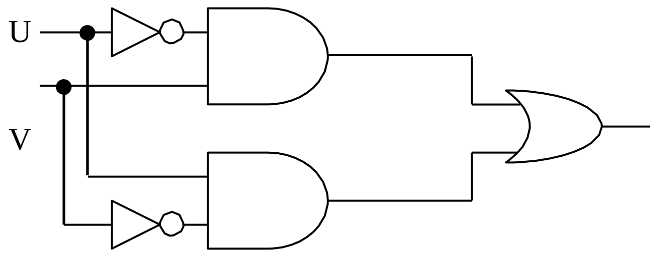
Short Answer Questions (2 marks)

Q1. Write the equivalent Boolean Expression for the following Logic Circuit.



Ans : F (P, Q) = (P' + Q) . (P + Q')

Q2. Write the equivalent Boolean Expression for the following Logic Circuit.



Ans: F (U , V) = U' . V + U . V'

Communication & Open Source Concepts

Very Short Answer Questions

Q1. What is a network?

Ans : A network is interconnected collection of autonomous computers that can share and exchange information.

Q2. Mention the advantages of networking.

Ans : The advantages of networking are Resource sharing, Reliability, Reduced cost, Fast communication.

Q3. Define server. Mention the types of servers.

Ans : A computer that facilitates the sharing of data, software, and hardware resources on the network is termed as server. The two types of servers are non-dedicated & dedicated servers.

Q4. What is NIU? What does the MAC address refer to?

Ans : A Network Interface Unit is an interpreter that helps establish communication between the server and workstations. The MAC address refers to the physical address assigned by NIC manufacturer.

Q5. What are the different types of switching techniques?

Ans : The different types of switching techniques are circuit switching, Message switching and packet switching.

Q6. What is a communication channel? Give examples of guided media & unguided media.

Ans : Communication channel means the connecting cables that link various workstations. The examples of guided media are Twisted-pair cables, Coaxial cables, Fibre-optics cables. The examples of unguided media are micro wave, radio wave, satellite, infrared wave etc.

Q7. Define the term Bandwidth. Give any one unit of Bandwidth.

Ans : Bandwidth is referred to the volume of information per unit of time that a transmission medium (like an Internet connection) can handle.

OR

The amount of data that can be transmitted in a fixed amount of time is known as bandwidth. For digital devices, the bandwidth is usually expressed in bits per second(bps) or bytes per second. For analog devices, the bandwidth is expressed in cycles per second, or Hertz (Hz).

Q8. Expand : HTTP, FTP, TCP/IP, SLIP, PPP, NCP, GSM, TDMA, CDMA, WLL, PSTN.

Ans : HTTP : Hypertext Transfer Protocol, FTP : File Transfer Protocol
TCP/IP : Transmission Control Protocol / Internet Protocol
SLIP : Serial Line Internet Protocol, PPP : Point to Point Protocol
NCP : Network Control Protocol, GSM : Global System for Mobile
TDMA : Time Division Multiple Access, CDMA : Code Division Multiple Access
WLL : Wireless Local Loop PSTN : Public Switched Telephone Network

Q9. What are the different types of networks?

Ans : The different types of networks are :

(a) Local Area Network (LAN): This computer network is confined to a localised area such as an office or a campus.

(b) Metropolitan Area Network (MAN): This is the network that links computer facilities within a city.

(c) Wide Area Network (WAN): This network spreads over large distances like across countries or even continents.

Q10. What do you mean by network topology? What are the most popular topologies?

Ans : Topology refers to the way in which the workstations attached to the network are interconnected. The most popular topologies are Bus Topology, Ring Topology, Star Topology, Tree Topology.

Q11. What are the advantages of E-mail?

Ans : The advantages of E-mail are Low cost, Speed, Waste reduction, Ease of use, Record maintenance, Patience.

Q12. What is WWW? What are the WWW attributes?

Ans : The World Wide Web (WWW) is a set of protocols that allow you to access any document on the NET through a naming system based on URLs. The various WWW attributes are : User-friendly, Multimedia documents, Hypertext and Hyperlinks, Interactive, Frames.

Q13. What is web browser? Give any two examples of web browsers.

Ans : A web browser is a WWW client that navigates through the WWW and displays web pages. Examples of web browsers are Internet Explorer, Netscape Navigator and Mosaic.

Q14. Define the terms: Web site, Web page, Home page, Web portal.

Ans : Web site : A location on a net server is called a web site.
Web page: A document that uses Hypertext Transfer Protocol is called a Web page.
Home page : The top-level web page of a website is called Home page.
Web portal : The website which hosts other websites is called Web portal.

Q15. When do you prefer XML over HTML and why?

Ans : The first benefit of XML is that because you are writing your own markup language, you are not restricted to a limited set of tags defined by proprietary vendors. Rather than waiting for standards bodies to adopt tag set enhancements (a process which can take quite some time), or for browser companies to adopt each other's standards, with XML, you can create your own set of tags at your own pace.

Q16. Compare authorisation with authentication.

Ans : Authorisation determines whether the service provider has granted access to the web service to the requestor. Authorisation is performed by asking the user a legal login id. Authentication ensures that each entity involved in using a web service is what actually claims to be. It is termed as password protection as the authorised user is asked to provide a valid password.

Q17. What is Firewall? How firewall protects our network?

Ans : The system designed to prevent unauthorised access to or from a private network is called firewall. It is a device or set of devices configured to permit, deny, encrypt, decrypt, or proxy all (in and out) computer traffic between different security domains based upon a set of rules and other criteria.

Q18. What are cookies?

Ans : Cookies are messages that a Web server transmits to a Web browser so that the Web server can keep track of the user's activity on a specific Website.

Q19. How is a cracker different from a hacker? If someone has hacked your Website, to whom you lodge the Complain?

Ans : A cracker is a malicious programmer who breaks into secure system whereas a hacker is more interested in gaining knowledge about computer systems. If someone has hacked our Website the complaint has to be lodged with the Police under IT Act.

Q20. What is Cyberlaw?

Ans : Cyberlaw is a generic term, which refers to all the legal and regulatory aspects of Internet and the World Wide Web.

Q21. Expand : (i) FLOSS (ii) FSF (iii) OSI (iv) W3C

Ans : (i) FLOSS = Free Libre and Open Source Software
(ii) FSF = Free Software Foundation (iii) OSI = Open Source Initiative
(iv) W3C = World Wide Web Consortium

Q22. What is a computer virus? Write the basic types of viruses.

Ans : Computer virus is a malicious program that requires a host and is designed to make a system sick. Broadly three types of viruses are File infectors, Boot Sector Viruses and Macro viruses.

Q23. Define the terms : Trojan Horses, Worm, Spam.

Ans : Trojan Horses : It is code hidden in a program such as a game or spreadsheet that looks safe to run but has hidden side effects.

Worm : A worm is a program designed to replicate.

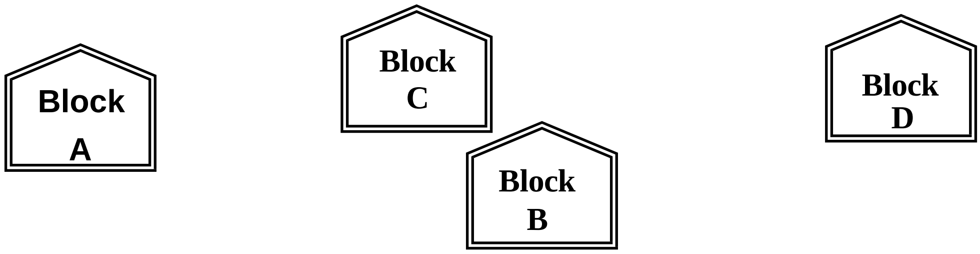
Spam : It refers to electronic junk mail or junk newsgroup posting.

Q24. What is cloud computing? Write any two features of it.

Ans. The practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer

Features : 1. On demand self service 2. Broad network access
3. Resource pooling 4. Rapid elasticity
5. Measured service 6. Non dependence on local hardware

Q1. Knowledge Supplement Organisation has set up its new centre at Mangalore for its office and web based activities. It has 4 blocks of buildings as shown in the diagram below:



Centre to centre distances between various blocks

Black A to Block B	50 m
Block B to Block C	150 m
Block C to Block D	25 m
Block A to Block D	170 m
Block B to Block D	125 m
Block A to Block C	90 m

Number of Computers

Black A	25
Block B	50
Block C	125
Block D	10

(a) Suggest a cable layout of connections between the blocks.

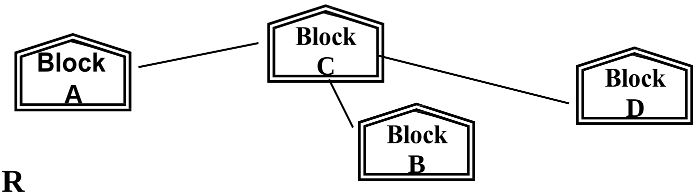
(b) Suggest the most suitable place (i.e. block) to house the server of this organisation with a suitable reason.

(c) Suggest the placement of the following devices with justification

(i) Repeater (ii) Hub/Switch

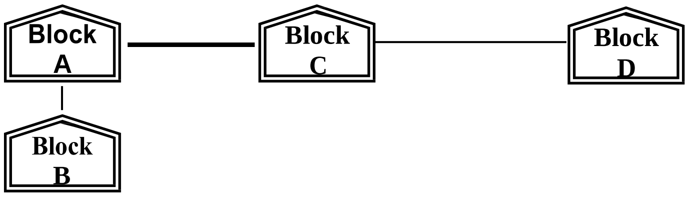
(d) The organization is planning to link its front office situated in the city in a hilly region where cable connection is not feasible, suggest an economic way to connect it with reasonably high speed?

Ans : (a)



Layout 1

OR



Layout 2

Since the distance between Block A and Block B is quite short

(b) The most suitable place / block to house the server of this organisation would be Block C, as this block contains the maximum number of computers, thus decreasing the cabling cost for most of the computers as well as increasing the efficiency of the maximum computers in the network.

(c) (i) For Layout 1, since the cabling distance between Blocks A and C, and that between B and C are quite large, so a repeater each, would ideally be needed along their path to avoid loss of signals during the course of data flow in these routes.

For layout 2, since the distance between Blocks A and C is large so a repeater would ideally be placed in between this path

(ii) A hub/switch each would be needed in all the blocks, to interconnect the group of cables from the different computers in each block.

(d) The most economic way to connect it with a reasonable high speed would be to use radio wave transmission, as they are easy to install, can travel long distances, and penetrate buildings easily, so they are widely used for communication, both indoors and outdoors. Radio waves also have the advantage of being omni directional, which is they can travel in all the directions from the source, so that the transmitter and receiver do not have to be carefully aligned physically