Vidyamandir Classes
Gurukul for IITJEE Preparation

CHAPTER - 1	Review of C++

1. What is the difference between 'a' and "a" in C++?

Sol. Characters enclosed in single quotes are character contains in C++. Thus 'a' is a character contant.
Characters enclosed in double quotes are string-literals which are array of characters. Thus "a" is a string i.e, in memory "a" is represented as "a\0" where \0 (null) character is the terminator of the string.
The size of 'a' is 1 character whereas the size of "a" is 2 characters.

2. What is a reference variable? What is its use?

Sol. 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.

3. What is the difference between 25L and 25?

Sol. An l(small L) or L suffix indicates it is long integer constant. Thus 25L is a long integer constant and 25 is an integer constant.

4. Arrange the following data types from smallest to largest : float, char, double, long double, long, short, int.

Sol. char, short, int, long, float, double, long double.

5. What is wrong with the following statement?
const int y ;

Sol. In the above statement, the constant has been declared but not initialized. Thus, the correct statement may be
const int y = 0; // (Any integer value can be stored in y).

6. How is integer 024 different from integer 24?

Sol. The integer 024 represents an octal number which is equivalent to 20 in decimal number system. The integer 24 represents number 24 in decimal number system which is equivalent to 030 in octal number system.

7. Which of the following two statements are valid? Why? Also write their results.
int x ;
(i) x = 1,024 ; (ii) x = (1,024)

Sol. (i) Invalid because an integer constant does not contain commas.
(ii) Valid. The x will be having value 024, the result of comma operator. (Left to right evaluation are choices compiler).

8. What will be result of following if ans = 6 initially?
(a) cout << ans = 8 ; (b) cout << ans == 8 ;

Sol. (a) 8 ; (b) 0 (which means false)

9. What is the similarity and difference between break and continue statements?

Sol. **Similarity :** Both break and continue are jump statements.
Difference : The break statement terminates the entire loop execution whereas continue statement terminates single pass of the loop and jumps to the next counter.

10. What is an inline function?

Sol. An inline function is that which is not invoked (i.e., loaded afresh) at the time of function call rather its code is replaced in the program at the place of function call during compilation. Thus, inline functions save overheads of a function call.

11. How is a static variable different from a static function?

Sol. A static variable has a function scope but global life time. A static function has a file scope instead of program scope.

12. Name the header file to be included for the use of following built – in functions :

(i)	frexp( )	(ii)	toupper( )	(iii)	getc( )	(iv)	strcat( )
(v)	isupper( )	(vi)	setw( )	(vii)	exp( )	(viii)	strcmp( )
(ix)	strcpy( )	(x)	isdigit( )	(xi)	log( )	(xii)	puts( )
(xiii)	strcat( )	(xiv)	scanf( )	(xv)	getchar( )	(xvi)	clrscr( )
(xvii)	strcpy( )	(xviii)	getxy( )	(xix)	puts( )	(xx)	gets( )
(xxi)	cos( )	(xxii)	setw( )	(xxiii)	toupper( )	(xxiv)	strcpy( )
(xxv)	isalnum( )	(xxvi)	gets( )	(xxvii)	fabs( )	(xxviii)	strlen( )

Sol.

(i)	math.h	(ii)	ctype.h	(iii)	stdio.h	(iv)	string.h
(v)	ctype.h	(vi)	iomanip.h	(vii)	math.h	(viii)	string.h
(ix)	string.h	(x)	ctype.h	(xi)	math.h	(xii)	stdio.h
(xiii)	string.h	(xiv)	stdio.h	(xv)	stdio.h	(xvi)	conio.h
(xvii)	string.h	(xviii)	conio.h	(xix)	stdio.h	(xx)	stdio.h
(xxi)	math.h	(xxii)	iomanip.h	(xxiii)	ctype.h	(xxiv)	string.h
(xxv)	ctype.h	(xxvi)	stdio.h	(xxvii)	math.h	(xxviii)	string.h

13. If value is an identifier of int type and is holding value 200, is the following statement correct?

char code = value ;

Sol. Yes, it is. This is because a character can hold characters with equivalent ASCII values in the range 0–255.

14. Explain the following : (a) Data abstraction (b) Encapsulation (c) Inheritance (d) Polymorphism

Sol. **(A) Data abstraction :**

It refers to the act of representing the essential features without including the background details or explanations. For example, in a 'switch board', you only press certain switches according to your requirement. What is happening inside, how it is happening, you needn't know. This abstraction.

(B) Encapsulation :

It is the wrapping up of data and functions (that operate on the data) into a single unit (called class). Encapsulation ensures that data of a class can be accessed only by authorized functions (member functions and friends functions of a class) . The data cannot be accessed directly, thus, it is safe from accidental alternation.

(C) Inheritance :

It refers to the capability of one class of things to inherit capabilities or properties from another class. For instance, the class 'car' inherits some of its properties from the class 'Automobiles' which inherits some of its properties from another class 'Vehicles'. The inheritance not only ensures the closeness with real world but also supports reusability of code.

(D) Polymorphism :

It is the property by which the same message can be sent to objects of several different classes, and each object can respond to it in a different way depending upon its class. For example, 6 + 9 results into 15 but 'X' + 'YZ' results into 'XYZ' . The same operator symbol '+' is able distinguish between the two operations (summation and concatenation) depending upon the data type it is working on.

15. How does OOPs overcome the shortcomings of traditional programming approaches?

Sol. The traditional programming approaches emphasize more on doing things. The algorithm for the solution plays a key role here. The data is not given as much concern as required. Data is after all, the reason for a program's existence.
Another problem associated with traditional programming languages is that they do no model the real world very well.
The OOPs concept overcomes these short comings by (i) the method of reusing the code through inheritance (ii) giving data the prime consideration and (iii) modelling the real world entities classes and objects. It is the data (and associated functions) that is the key factor while declaring classes. And the real world behavior is reflected through objects and their properties.

16. What are the advantages offered by inheritance?

Sol. The major advantages offered by inheritance are :

1. Its capability to express the inheritance relationship which makes it ensure the closeness with the real – world models

2. Inheritance supports reusability of code. It allows the addition of additional features to an existing class without modifying it ; one can derive a new class (sub-class) from an existing one and add new features to it.

3. Inheritance it transitive in nature i.e., a class B inherits properties of another class A, then all subclasses of B will automatically inherit the properties of A Transitivity carries a major benefit with it – any change in a class is automatically reflected across all classes that inherit from it.

17. Encapsulation is one of the major properties of OOP. How is it implemented in C++?

Sol. Encapsulation is a way to implement data hiding by wrapping up data and associated functions into a single unit, class. The class groups its member (data and functions) into three sections : private, protected and public.
A class binds together data and its associated functions under one unit thereby enforcing encapsulation as encapsulation means wrapping up data and associated functions together into a single unit.

18. How are abstraction and encapsulation inter-related?

Sol. Encapsulation wraps up data and functions under single unit and ensures that only essential features get represented without representing the background details which is nothing but Abstraction. Therefore, encapsulation is a way of implementing abstraction.

19. What is a base class? What is a derived class? How are these two interrelated?

Sol. A derived class is a class that inherits properties from some other class. It is also known as sub class.
A base class is a class whose properties are inherited by its derived class. It is also known as super class
A derived class inherits properties from base class reverse is not true.

20. What is the difference between a keyword and an identifier?

Sol. **Keywords** is a special word, that has a special meaning and purpose. Keywords are reserved and are a few. For example, **goto, switch, else** etc. are keywords in C++.
Identifier is the user-defined name given to a part of a program viz. variable, object, function etc. Identifiers are not reserved. These are defined by the user and they can have letters, digits and a symbol underscore. They must begin with either a letter or underscore. For instance, _chk, chess, trial etc. are valid identifiers in C++.

21. Why is main function special? Give two reasons.

Sol. Whenever a C++ program is executed only the **main( )** is executed i.e., execution of the program starts and at **main( )** The **main ( )** is the driver function of the program. If it is not present in a program, no execution can take place.

22. Write two advantages of using include compiler directive.

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

(ii) It supports modularity i.e., a bigger program can be decomposed in terms of header files and later included through this directive.

23. Explain the function and usage of a union giving an example.

Sol. A union is a memory location that is shared by two or more different variables, generally of different types at different times.

Following declaration declares a union share having two variables (integer and character type) and creates a union object cnvt of union type share :

```
union share {
    int i ;
    char ch ;
    };
share cnvt;
```

In the union cnvt, both integer i can character **ch** share the same memory location. (of curse, i (being integer) occupies 2 bytes and **ch** (being character) uses only 1 byte).

At any point, you can refer to the data stored in a cnvt as either an integer (as cnvt.i) or character (as cnvt.ch).

24. What is a compiler directive? Why do we need #include in a c++ program? Name the include file, to which following built–in functions belong to:

(a)	strcmp()	(b)	randomize()	(c)	setw()
(d)	isalnum()	(e)	sin()	(f)	gotoxy()

Sol. A compiler directive is an instruction for the compiler to carry out a specific job.

To include a specific header file (its declarations, routines, macros etc) in a particular program #include directive is used. It instructs the compiler to read a specific file and include that file under the current file. After including the specified file, the compiler compiles the total code (i.e., code of included file as well as the current file).

(a)	strcmp() is contained in string.h	(b)	randomize() is contained in stdlib.h
(c)	setw() is contained in iomanip.h	(d)	isalnum() is contained in ctype.h
(e)	sin() is contained in math.h	(f)	gotoxy() is contained in conio.h.

25. What is the difference between fundamental data types and derived data types? Explain with examples.

Sol. Fundamental (or atomic) data types are those data types which are not composed of other data types. There are five fundamental data types in C++ viz. char, int, float, double and void.

The derived data types are those data types which are composed of fundamental data types. Some examples are : classes, structures, unions, enumerations.

26. Give the difference between the type casting and automatic type conversion. Also, give a suitable C++ code to illustrate both.

Sol. Explicit Type Casting is used by the programmer to convert value of one type to another type. It is forced type conversion. It is done by putting the target data type in parentheses before the data to be converted, for example, in the following statement 15 would be type cast into 15.0 first.

```
float x = (float) 15/4 ; // 3.75 will be assigned as result.
```

Automatic Type Conversion is the type conversion done by the compiler itself wherever required. It is implicit type conversion, e.g., in the following code before assigning the value 3 to **float x,** it will be automatically converted to 3.0.

```
float x = 3;
```

27. Differentiable between a Run Time Error and Syntax Error? Also give suitable examples of each in C++.

Sol. Run Time Error refers to an error that occurs during execution of program i.e., during run time generally when some resource is unavailable e.g., “File not found”. “Out of Memory” etc. are runtime errors.

Syntax Error refers to error resulting from violation of programming language rules. e.g., a * b = c is a syntax error. Similarly **void main[]** is also a syntax error.

28. Differentiable between a Logical Error and Syntax Error? Also give suitable examples of each in C++.

Sol. Logical Error is an error which occurs because of wrong interpretation of logic. With logical error(s), the code is syntactically correct but does something undesired. For example, if in place of c = a + b; if by mistake, c = a * b ; is written, it will be logical error.

A syntax error is the error that occurs when statements are wrongly written violating rules of the programming language. For example, MAX + 2 = DMAX is a syntax error as an expression cannot appear on the left side of an assignment operator.

29. What is the difference between Actual Parameter and Formal Parameter? Also, give a suitable C++ code to illustrate both.

Sol. Actual Parameter is a parameter, which is used in function call statement to send the value from calling function to the called function.

Formal Parameter is a parameter, which is used in function header of the called function to receive the value form actual parameter.

For example,

```
void Multiply(int T)     //T is formal parameter
{
    cout<< 5*t;
}
int main()
{
   int A = 45;
   Multiply(A);          //A is actual parameter sent to Multiply()
   return 0;
}
```

30. What is the difference between call by value and call by reference in a user defined function in C++? Give an example to illustrate the same.

Or

What is the difference between call by value and call by reference? Also, given a suitable C++ code to illustrate both.

Or

What is the difference between call by value and call by reference? Given an example in C++ to illustrate both.

Sol. In call by value method, the called function creates its own copies of the original values sent to it. Any changes, that are made, occur on the called function's copy of values and are not reflected back to the calling function.

In call by reference method, the called function accesses and works with the original values using their references. Any changes, that occur, take place on the original values and are reflected back to the calling code.

Example of call by value

```
int main()
{    int a = 5;
     cout << "a = " <<a;
    change(a);
   cout<<"\n a=" <<a;
   return 0;
}
void change (int b)
{    b=10;
   }
```

Output will be :

a = 5

a = 5

Example of call by reference

```
int main()
{
   int a = 5;
   cout<<"a= " <<a;
   change (a);
   cout<< "\n a = " << a;
   return 0;
}
void change (int & b)
   {   b = 10;
   }
```

Output will be :

a = 5

a = 10

31. What is the purpose of using a typedef command in C++? Explain with suitable example.

Sol. A typedef command defines a new name or an alias name or an alias name for an existing type. For example, all transactions generally involved amounts which are of (say) double type. So, for a bank application, we can safely provide an alias name as Amount to the predefined double type. For this, we shall write :

```
typedef double Amount ;
```

Now we can define any amount using the datatype Amount as :

Amount loan, balance, instalment, interest ;

32. What do you understand by default arguments and constant arguments? Write a short note on their usefulness.

Sol. The default argument is a way to specify predefined/default value to an argument in case it is not provided in the function call statement.

The const argument is a way to ensure that the argument declared as const cannot be altered in the function body. For example,

int F1(int time const, float prin=6000);

With above prototype, function calls like this:

int t = 5 ;

int r = F1(t);

will consider the default value 6000 for second argument as it is missing in the function call statement further first argument cannot be modified in F1() i.e.,

```
int F1 (int time const, float prin = 6000)
{       time = 10;              // → NOT ALLOWED
          …….
}
```

33. Define a macro. How is it different from a constant defined through const ?

Sol. A macro refers, to #define definition that is used to define words to enhance readability and understandability. Before compilation the preprocessor replaces every occurrence of macro as per the definition e.g.,

#define EOF 0

will replace every occurrence of EOF with 0 within program.

A macro is a preprocessor directive and is text–replaced during compilation whereas a const is a type identifier having constant value. Other major differences between a macro and const are :

(i) A const object is subject to scoping rules for variables whereas a #define is not.

(ii) The compiler does not type–check a macro whereas a const is definitely type-checked.

34. Which of the following are correctly formed #define statements?

(i) #define INCH PER FEET 12

(ii) #define DOUBLE (X) (2*X)

(iii) #define DOUBLE(X) 2*X

(iv) #define DOUBLE(X) (2*X)

Name the macros that get defined through above statements.

Sol. All of the above statements are syntactically correct but they may pose following errors :

(i) It will define a macro **INCH** with value to be substituted as **PER FEET 12,** which may result into errors in expressions.

(ii) It will define a macro namely **DOUBLE** having substitute value as **(X) (2*X)** (because of a space following DOUBLE).

(iii) It will define a macro **DOUBLE(X)** with X as parameter having substitute value as **2*X.**

(iv) It will define a macro **DOUBLE(X)** with X as parameter, having substitute value as **(2*X).**

35. Find errors if any in the following program;

```
// the aim of this program is to interchange the values of two variables by using
// the function swap(float, float)
#include<iostream.h>
void swap(float, float);
voidmain(
    float a, b;
    cout<<"\nEnter a real no. x";
    cin>>x;
    cout<<"Enter a real no. y";
    cin>>y;
    swap(x,y);
    cout<<"\n Now the value of x = "<<x;
    cout<<"\n Now the value of y = "<<y;
    }
    void swap(float a, float b)
    {
        int temp;
        temp = a;
        b = a;
        a = temp;
        }
```

Sol. The function swap() intends to swap the values of two passed parameters. However, values are passed to it by value, which means any changes in the parameters will not be reflected back. Therefore, the actual purpose of the function is defeated. To correct this problem, the parameters should be passed by reference i.e. the prototype of **swap()** should be

```
void swap (float & , float &);
```

and the function header should be

```
void swap (float & a, float & b)
⋮
```

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

```
#include [isotream.h]
typedef char Text (80);
void main()
{
   Text T = "Indian"
   int Count = strlen(T)
   cout<<T<<'has' <<Count<<'characters'<<endl;
   }
```

Sol.

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

37. Rewrite the following program after removing the syntactical error(s), if any. Underline each correction.

```
#include<iostream.h>
const int Max 10;
void main()
{
int Numbers[Max];
Numbers = {20, 50, 10, 30, 40};
for (Loc = Max – 1; Loc > =0; Loc– –)
cout >> Numbers[Loc];
}
```

Sol.

```
#include<stdlib.h>
const int Max = 10;
void main()
{
int Numbers[Max] = {20, 50, 10, 30, 40};
for(int Loc = Max – 1; Loc > = 0; Loc – –)
cout << Numbers[Loc];
}
```

38. Rewrite the corrected code for the following program. Underline each correction (if any);

```
#include<iostream.h>
structure Swimmingclub
{
      int memnumber;
      char memname[20];
      char memtype[] = "LIG";
};
void main()
{
```

```
Swimmingclub per1, per2;
cin >> memnumber.per1;
cout<<"Member Name:";
cin>>per1.membername;
per1.memtype = "HIG";
per2 = per1;
cin << "Member Number:" << per2.member;
cin<<"Member Name"<<per2.memname;
cin<<"Member Number;"<<per2.memtype;
}
```

Sol.

```
#include<iostream.h>
#include<string.h>
struct Swimmingclub                  //should be struct
{
     int memnumber;                 //a variable name cannot contain spaces
     char memname[20];               //cannot initialize a structure member
     char memtype[4];               //inside structure definition
};
void main()
{
   Swimmingclub per1, per 2;
   cout << "Member Number";          //should be cout
   cin >> per1. memnumber;           //structure.member
   cout << "Member Name:" ;
   cin>>per1.memname;                //member name is not the structure member
   strcy (per1. memtype, "HIG");      //to copy string, strcpy is used
   per2 = per1 ;
   cout << "Member Number:" << per.memnumber;     //should be cout
   cout << "Member Name" << per2.memname;         //should be cout
   cout << "Member type" << per2. memtype;
}
```

39. Rewrite the following program after removing all the syntax error(s), if any.

```
#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 0 :
        case 2 : cout << Y*X[I] << endl;  break;
        case 1 :
        case 3 : cout >> Y + X[I];
        }
   }
```

Sol.
```
#include<iostream.h>
void main()
{int X[] = {60, 50, 30, 40}, Y, Count = 4;
cin >> Y;
for (int I = Count -1; I > = 0; I --)
switch (I)
{
    case 0 :
    case 2 : cout << Y* X[I] << endl;    break;
    case 1 :
    case 3 : cout << Y + X[I];
}
}
```

40. Find the syntax error(s), if any, in the following program :
```
#include<iostream.h>
main()
{
int x[5], *y, z[5];
for (i = 0 ; i < 5; i++)
{       x [i] = i;
        z[i] = i + 3;
        y = z;
        x = y;
        }
}
```

Sol.

	Erroneous Statements	Errors and Corrections
1.	int x[5], *y, z[5];	There should not be space between array name and its size as it is in x and [5]. The corrected code is : int x[5], *y, z[5];
2.	for(i = 0; i < 5; i++)	i variable not defined. The corrected code shall be for (int i = 0; i < 5; i++)
3.	x = y;	x being array name can not be assigned any other value. Thus x = y is erroneous. The corrected code could be x[i]=*y; or y = x ;

41. Find the syntax error(s), if any, in the following program :
```
include<iostream.h>
void main()
{
  intR ; W = 90;
  while ; W > 60
  {
     R = W - 50;
     switch(W)
     {
     20 : cout << "Lower Range"<<endl;
```

```
30 : cout << "Middle Range"<<endl;
20: cout << "Higher Range" << endl;
}
}
}
```

Sol. :

	Given code	Errors and Corrections
1.	include <iostream.h>	It should be #include <iostream.h>
2.	void main()	
3.	{	
4.	int R; W = 90;	Variable list should be separated using commas. It should be: int R, W = 90;
5.	while W > 60	Test expression should be in braces as shown below: while (W > 60)
6.	{	
7.	R = W – 50;	
8.	switch(W)	
9.	{	
10.	20 : cout<<"Lower Range"<<endl;	It should be case 20 and the text should be in " " not in " ". The correct statement would be: case 20 : cout << "Lower Range" <<endl;
11.	30 ; cout<< "Middle Range"<< endl;	Same Error, as step 10. The correct statement would be : case 30 : cout << "Middle Range" << endl;
12.	20 : cout <<"Higher Range" <<endl;	Two cases cannot have same value for comparison. Also keyword case is missing. The correct statement would be : case 40 : cout << "Higher Range" << endl;
13	}	
14	}	
15	}	

42. What will be the output of the following code fragment

```
cout<<"Enter a value";
int a;
cin >>a;
if (a = 5)
cout << "Five"
else
cout <<"Not Five"
```

if the input given is (i) 7 (ii) 5?

Sol. (i) Five (ii) Five (Reason : use of = in place of ==)

43. Please list the output of the program below. In other words, every time there is a cout statement, list the values that will appear on the screen.

```
#include<iostream.h>
int main(void)
{
   for(int i = 1; i<3 ; i++)
         for (int j = 1; j <= 5 ; j += 2)
      cout << i <<"*" <<j<<"=" <<i * j <<endl ;
      double xx = 1;
      while (xx << 32)
      xx *=2 ;
      cout << "xx =" << xx << endl;
      int i = 8, j = 3;
      int k = i/j;
      double x = i/j;
      double y = ((double) i/(double) j);
      cout <<"Now k = " <<k <<",x="<<x <<", y = "<< y << endl;
      return 0;
      }
```

Sol.
1 * 1 = 1
1 * 3 = 3
1 * 5 = 5
2 * 1 = 2
2 * 3 = 6
2 * 5 = 10
xx = 32
Now k = 2, x = 2.0, y = 2.666.

44. What is the output of the following program.

```
#include<iostream.h>
void main()
{        struct point
         {     int x, y;
         }polygon[] = {{1,2], {1,4}, {2, 4}, {2,2}};
      point *a;
      a = polygon;
      a++ ;
      a -> x++;
      cout << polygon -> x <<endl;
}
```

Sol. Polygon points to polygon[0] and polygon[0].x i.e., polygon–> x is 1. (a –>x will represent 2 as a++ (0 + 1) points to polygon[1] and a ->x++ makes polygon[1].x as 1 + 1 = 2).

45. Consider the following program:

```
#define two(x) x * x
# define ddouble(x) x + x
main()
{
```

```
int num, sum, product;
num = 1;
sum = --two(num); --sum;
product = -- ddouble(num);
cout<<sum<<product;
}
```

Predict the output of the above program

Sol. - 1 - 2

Because, expression sum = – –two(num), will expand to – – num*num resulting in sum being 0 and num being 0. Then – – sum will make sum as – 1.

Expression product = – – ddouble (num) will expand to – – num + num resulting in –2 i.e., (–1 + (–1)) as num was earlier 0.

46. Find the output of the following program.

```
#include<iostream.h>
#include<ctype.h>
typedef char Txt80[80];
void main()
{
   char *PText;
   Txt80 Txt= "Ur2GReAt";
   int N = 6;
   PText = Txt;
   while(N>=3)
   {
   Txt[N] = (isupper(Txt[N]) ? tolower(Txt[N]) : toupper(Txt[N]));
   cout << PText << endl;
   N--;
   PText++;
   }
   }
```

Sol. :

```
Ur2GReat
r2GREat
2GrEat
grEat
```

47. Find the output of the following program :

```
#include<iostream.h>
void main()
{
   int U = 10, V = 20;
   for (int I = 1; I < = 2; I++)
   {
      cout << "[1]" << U++ << "&" << V - 5 << endl;
      cout << "[2]" << ++V << "&" << U + 2 << endl;
      }        }
```

Sol.

```
[1]     10      &       15
[2]     21      &       13
[1]     11      &       16
[2]     22      &       14
```

48. Find the output of the following program :

```
void main();
{
   int x = 5, y = 5;
   cout << x ++ ;
   cout << ",";
   cout <<++x;
   cout << ",";
   cout << y++<<"," <<++y;
}
```

Sol. 5, 7, 6, 6

49. Give the output of the following program.

```
void main()
{
   char *p = "Difficult":
   char c;
   c = ++*p++ ;
   printf{"%c",c};
   }
```

Sol. E

50. Give the output of the following program.

```
void main()
{
   int i = 3;
   prinf("%d%d%d", i + 5, i - 4, i);
}
```

Sol. : 8 –1 3

51. How many way are there in C++ to represent an integer constant ?

Sol. In C++, an integer constant can be represented in three ways :

1. As a decimal integer constant in which the integer constant consists of a sequence of digits but it does not begin with digit 0 (zero). For example, 1234 is a decimal integer constant but 01234 is not a decimal integer constant.
2. As an octal integer constant in which the integer constant begins with digit 0 (zero). For example, 9(in decimal) can be written as 011(octal equivalent of 9) as octal integer. (In C++, 08, 09, 018, 019, 028.. etc. are not valid integers as 0 in the beginning means these are octal and octal number system recognizes symbols 0-7 only.
3. As a hexadecimal integer constant in which the integer begins with 0x or 0X. For instance, 11 (in decimal) can be written as 0XB (hexadecimal equivalent of 11).

52. Find the output of following program :

```
#include<iostream.h?
struck STOCK
{
    int Ino, Qty:
    }
    void Buy (STOCK &I, int TQ = 2)
    {
       I.Qty += TQ;
       void main()
       STOCK I[2] = { { 101, 50}, {103, 20}};
```

```
    Buy(I[1], 5);
    cout<<I[1].Ino<<":" <<I[1].Qty<<endl;
    Buy(I[0],10);
    cout<<I[0].Ino<<":"<<I[0].Qty<<endl;
    Buy(I[1]);
    cout<<I[1].Ino<<":"<<I[1].Qty<<endl;
    }
```

Sol. 103 : 25
101 : 60
103 : 27

52. Give the output of the following program :

```
#include<iostream.h>
int a = 3;
void demo(int x, int y, int &z)
{
   a += x + y;
   z = a + y;
   y += x;
   cout << x <<y<<z<<endl;
   }
   void main()
   {
      int a = 2, b = 5;
      demo (::a, a, b);
      cout << ::a << a << b << endl;
      demo(::a, a, b);
      }
```

Sol. :
3 5 10
8 2 10
8 10 20

Chapter - 2 | Objected Oriented Programming in C++

1. What is abstract class ?

Sol. An abstract class is a class that defines an interface, but does not necessarily provide implementations for all its member functions. No objects of an abstract class exist i. e., it can not be instantiated.

2. What is concrete class ?

Sol. A concrete class is a derived class of an abstract class that implements all the missing functionality. A concrete class can be instantiated i.e., its objects can be created.

3. What is function overloading ?

Sol. A function name having several definitions that are differentiable by the number or types of their arguments is known as an overloaded function and this process is known as function overloading.

4. What is the importance of function overloading?

Sol. Function overloading not only implements polymorphism but also reduces number of comparisons in a program and thereby makes the program run faster.

5. What are inline functions ?

Sol. Inline functions are short functions that are not actually called; rather, their code is expanded (inlne) at the point of function call.

6. What is a constructor?

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

7. What is a destructor?

Sol. A destructor is also a member function whose name is the same as the class name but is preceded by tilde (~). A destructor destroys the objects that have been created by a constructor. It destroys the values of the object being destroyed. (It is called when the object goes out of scope)

8. Distinguish between the following two statements :
time T1(13, 10, 25); //statement 1
time T1 = time(13, 10, 25) ; //statement 2.

Sol. The first statement creates the object T1 by calling the time constructor implicitly. On the other hand, the second statement creates the object T1 by calling the time constructor explicitly.

9. How does a class enforce data – hiding, abstraction and encapsulation ?

Sol. A class binds together data and its associated functions under one unit thereby enforcing encapsulation as encapsulation means wrapping up data and associated functions together into a single unit.
A class groups its members into there sections : private, protected, and public. The private and protected members remain hidden from outside world. Thus through private and protected members, a class enforces data – hiding.
The outside world is given only the essential and necessary information through public members, rest of the things remain hidden, which is nothing but abstraction. As abstraction means representation of essential features without including the background details and explanation.

10. What do you understand by Data Encapsulatin and Data Hiding ? Also given a suitable C++ code to illustrate both.

Or

Define the term Data Hiding in the context of Object Oriented Programming. Give a suitable example using a C++ code to illustrate the same.

Or

Define the term Data Encapsulation in the context of Object Oriented Programming. Give a suitable example using a C++ code to illustrate the same.

Sol. Data Encapsulation is the wrapping up of data and operations/functions (that operate on the data) into single unit (called class) is known as Encapsulation.

Data Hiding is keeping un-essential and critical data hidden from outside world to prevent any accidental charges or unsafe practices.

A Class implements both data encapsulation and data hiding e.g.,

```
class clock
{       double alarmtime;
        char brand[50];
        double time;            (Hidden data because it is private.)
public ;
void getTime();
void setTime();
void setAlarm();
void ringAlarm();
};              (Class implements encapsulation by defining data and functions as one unit.)
```

11. What do you understand by Polymorphism ? Give a suitable example of the same.

Sol. Polymorphism is the property by which the same message can be sent to objects of several different classes and each object can respond to it in a different way depending upon its class. In C++, Polymorphism is implemented through overloading (and virtual functions).

A function name having several definitions that are differentiable by the number and types of their arguments, is known as function overloading.

For example,

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

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

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

13. When will you make a function inline and why?

Sol. A function is made inline when the following features are there :

(i) the function is small.

(ii) the function is not returning any value and contains no return statement.

(iii) the function does not contain a loop or a switch or a goto.

(iv) the function does not contain static variables.

(v) the function is not recursive.

The inline functions run a little faster than the normal functions as function–calling overheads are saved.

14. While defining a class, you can define its methods (member functions) inside or outside the class definition. How are these two definitions different?

Sol. There are basically two differences in these two definitions :

1. In the why the function names are written. When a member function is defined outside the class definition, its name must be a qualified name i.e., its name must be in the form classname::function name whereas this is not required for an inside definition.
2. In the way the functions are processed. A member function defined inside the class definition are automatically trerated as inline functions which is not the case when the functions are defined outside the class definition unless the functions are explicitly made inline.

15. What is the significance of access labels in a class ?

Sol. A class provides three access labels namely, private, protected, and public.

A member declared as private or protected remains hidden from outside world and it can only be accessed by the member functions and friend functions of the class.

A member declared as public is made available to the outside world. That is, it can be accessed by any function, any expression in the program but only by using an object of the same class type.

These access labels enforce data – hiding and abstraction, OOP concepts.

16. What are the advantages and disadvantages of inline functions?

Sol. The main advantage of inline functions is that they save on the overheads of a function call as the function is not invoked, rather its code is replaced in the program.

The major disadvantage of inline functions is that with more function calls, more memory is wasted as for every function call, the same function code is inserted in the program. Repeated occurrences of same function code waste memory space.

17. What do you understand by a default constructor? What is its role? How is it equivalent to a constructor having default arguments?

Sol. A default constructor is the one that takes no arguments. It is automatically invoked when an object is created without providing any initial values. In case, the programmer has not defined a default constructor, the compiler automatically generates it.

It is equivalent to a constructor having default arguments because when no initial values are provided for an object at the time of its declaration, the constructor having default argument values is invoked same way as the default constructor is invoked. Therefore, these two are considered equivalent.

18. What do you mean by a temporary instance of a class ? What is the use ? How is it created ?

Sol. A temporary instance of a class means an anonymous object (object having no name) of the same class and which is shortlived. Its benefit it when an object is required only for very short time (say for an expression or a statement), we need not reserve memory for it for long. A temporary object for the same purpose can be created which remains in the memory as long as the statement defining it is getting executed, after the statement, this object is automatically destroyed and memory is released. Therefore, memory remains occupied only for the time when it is needed. A temporary instance is created by explicit call to the constructor. For instance, following statement creates a temporary instance of time type and invokes print() member function of time class for it.

time(10, 12, 30).print();

19. Discuss the various situations when a copy constructor is automatically invoked.

Sol. Generally, in the following three situations, a copy constructor in invoked.

1. When an object is defined and initialized with another object of the same type, then the copy constructor is invoked to copy the data values of one object to the other.

2. When an object is passed by value to a function, then the copy of the passed object is created for the function by invoking the copy constructor.

3. When a function returns an object, the copy constructor creates a temporary object to hold the return value of the function returning an object.

20. How many times is the copy constructor called in the following code?

```
Sample func(Sample u)
{
Sample v(u);
Sample w = v ;
return w ;
}
void main()
{
Sample x ;
Sample y = func(x);
Sample z = func(y);
}
```

Sol. : The copy constructor is called 8 times in this code. Each call to the function func() requires call to the copy constructor :

(i) When the parameter is passed by value by u

(ii) when v is initialized

(iii) when w is initialized

(iv) when w is returned.

21. Differentiate between Constructor and Destructor function with respect to Object Oriented Programming

Or

What is the use of a constructor function in a class ? Give a suitable example of a constructor function in a class.

Sol. A constructor is a member function of a class with these characteristics : (i) same name as that of the class, (ii) no return type. A Constructor is automatically invoked when an object is created and it is used to initialize data members of the object. Constructors can be overloaded.

A destructor is also a member function of a class with these characteristics : (i) same name as that of class with a ~ before the name, (ii) no return type. Destructors are automatically invoked when an object goes out of scope and it is used to free all resources to the object during run-time. Destructor cannot be overloaded.

22. What do you understand by default constructor and copy constructor functions used in classes ? How are these functions different from normal constructors?

Sol. A constructor that accepts no parameters is called default constructor.

A constructor that constructs an object by initializing it with another object of the same class is called copy constructor.

Normally, constructors can be default, parameterized or copy constructors. A parameterized constructor can take arguments and if its arguments have default values, it becomes equivalent to default constructors.

23. What do you understand about a base class and derived class ? If a base class and a derived class each include a member function with the same name and arguments, which member function will be called by the object of the derived class if the scope operator is not used?

Sol. A base case is the class whose properties are inherited by another class.

The derived class is the class that inherits properties from base class.

If a base class and a derived class each include a member function with the same name and argument, the member function of derived class will be called if the scope operator is not used.

24. How does inheritance influence the working of constructors and destructors?

Sol. Inheritance means a class (derived class) is inheriting properties form another class (the base class). When an object of the derived class is declared, in order to create it, firstly the constructor of the base class is invoked and then, the constructor of the derived class is invoked.

On the other hand when an object of the derived class is destroyed, first the destructor of the derived class is invoked followed by the destructor of the base class. That is, the destructors are invoked in the reverse order of constructor invocation.

25. What is containership ? How does it differ from inheritance ?

Sol. When a class contains objects of another class as its members, this kind of relationship is called containership or nesting.

Inheritance lets you create or define a specialized instance of a class that shares the properties of the class and at the same time adds new features to it. 'Containership', on the other hand, does not do any such job rather it facilitates to enclose objects of other classes inside a class. It is just a way to define an object which itself is collection of objects of other classes.

26. What do you understand by visibility modes in class derivations? What are these modes

Sol. Visibility modes determine the access specifiers to be, for the inheritable members of the base class in the derived class.

There are three visibility modes ; (i) public (ii) protected (iii) private

public visibility mode retains the original access specifiers of the inherited members in the derived class.

protected and private visibility modes makes the access specifier of inherited members as protected and private respectively in the derived class.

27. How does the invocation of constructors differ in derivation of class and nesting of classes?

Sol. A derived class invokes its base class's constructor explicitly i.e., by naming the constructor name while invoking it.

A container class invokes it member object's class constructor implicitly i.e., it does not name the constructor but through the object name it gets invoked.

For example,

```
case A  {
              ⋮
        };
case B  {
              ⋮
        };
class C : public A
        {       B object 1;
                    ⋮
        public:
              C( ) : A( ), object ( )
        {
                        ⋮
        }
        }
```

28. When does ambiguity arise in multiple inheritance? How can one resolve it ? What are virtual base classes ? What is their significance ?

Sol. : Ambiguity arises in an inheritance hierarchy of multiple, multilevel inheritance if an inherited member of common origin gets inherited via two different path.

This ambiguity is resolved through virtual base classes which ensures that only single copy of common origin gets inherited. When two or more objects are derived from a common base class, you can prevent multiple copies of base class from being present in an object derived from those objects by declaring the base class as virtual when it is inherited. This is accomplished by putting the keyword virtual before the base class name when it is inherited.

Example:

```
class base
{       public :
                int a ;
} ;
class D1 : virtual public base
{       public :
                int b ;
}
class D2 : virtual public base
{       public :
                int c ;
}
class D3 : public D1, public D2
{       public :
                int d ;
                void show ( )
                {
                        cout << a << b << c << d ;
                }
} ;
void main ( )
{       D3 Obj ;
        Obj . a = 10
        Obj . b = 20
        Obj . c = 30
        Obj . d = 40
        cout << Obj. a <<Obj.b<<Obj.c <<Obj.d ;    // Obj. show ( ) ;
}
```

29. Rewrite the following program after removing the syntactal error (if any). Underline each correction.

```
#include<iostream.h>
class BOOK
   {  long BId, Qty ;
   Public :
      void Purchase() {cin >> BId >> Qty; }
      void Sale
```

```
{    cout << setw(5) << BId << "Old: " << Qty<<endl;
   cout << "New: " <<--Qty << endl;
   }
   };
      void main()
      { BOOK B;
      B.Purchase();
      Sale();
      B.Sale()
}
```

Sol.

```
#include<iostream.h>
#include<iomanip.h>
class Book
{
 long BId, Qty;
    public :
        void Purchase(){cin >> BId >> Qty;}
        void Sale
        {cout << setw(5) << BId << "Old:" << Qty <, endl;
        cout <, "New:" << -- Qty << endl;
        }
        };
        void main()
        {
        Book B;
        B. Purchase();
        B.Sale();
        B.Sale();
        }
```

30. How is matching done in case of overloaded functions ?

Sol.: With a function call that uses the name of an overloaded function, the compiler follows the following steps in order to find the best match :

1. Search for an exact match is performed. If an exact match is found, the function is invoked.
2. If an exact match is not found, a match through promotion is searched for. Promotion means conversion of integer type char, short, enumeration and int into int or unsigned int (whatever is capable of representing all the values of the datatype being promoted) and conversion of float into double.
3. If first two steps fail then a match through application of C++ standard conversion rules is searched for.
4. If all the above mentioned steps fail, a match through application of user – defined conversions and build – in conversions is searched for.
5. If all these steps fail, the complier flashes a message reporting no match.
6. If more than one definition qualify for the function call, the compiler reports an ambiguous match.

31. Given the following code fragment :

```
        int x, b;
class X
{
    int x;
    float y;
```

```
char z ;
void int (void)
{x = y = z = 0;
}
public:
int a;
int b;
void sqr(int i)
{
    cout <<"Private sum:"<<(x + y) * 1;
    cout << "\n" << "Public sum:" <, (a + b) * 1 << "\n";
    }
    void start(int i, float j, char k);
    friend void sum(void)
    };    //class X
    void X :: start(in i, float j, char k)
    {
        int();
        x = i;
        y = j;
        z = k;
        }
        void sum(void)
        {cout << "sum=" << x + y + a + b << "\n";
        }
        X O1;
        void check(void)
        {int a = 10;
        cout << a << O1.b << x ;
        }
```

What all variables can be accessed by the following functions?
sqr ( ), start( ), sum (), check( )

Sol. sqr( ) can access X :: x, :: x, X :: y, X :: z, X :: a, X :: b, :: b
start( ) can access all the variables as that of sqr( )
sum( ) can also access all the variables as that of above two functions.
check( ) can access :: x, :: b, X :: a, X :: b and its local variable a.

32. Consider the following code fragment :

Sol.

```
int x = 5;
int y = 3;
class Outer
{   public:
    int x ;
    int a ;
    static int s ;
    class inner
    {int a;
    public:
        void f(int i)
        { x = i;
        s = i;
        y = i ;
```

```
a = i;
}
};          //Inner definition over
Inner I1;     //Inner object
};
void g(int i)
{   x = i;
    y = 1;
    a = i;
    s = i;
    }
    };      //Outer definition over
    Outer O1   //Outer object
    int main()
    {O1.I1.f(3);        //statement1
    O1.g(8):            //statement2
    return 0;
    }
```

What will be the values of :: x, ::y, Outer :: x, Outer ::a, Outer::s, Inner::a after statement 1 and statement 2 of above code?

Sol. After statement 1
:: x = 5; ::y = 3; Outer :: x = 3; Outer ::a = Not Defined; Outer :: s = 3; Inner :: a = 3
After statement 2
::x = 5 ; ::y = 8; Outer :: x = 8 ; Outer :: a = 8 ; Outer :: s = 8 ; Inner :: a = Its previous value

33. What will be output of following program :

```
# include<iostream.h>
# Include<conio.h>      // for cirscr( )
class X {
    int x ;
    float y;
    public :
    void init(void)
    {
        x = 0;
        y = 0 ;
        }
        void getval(int i, float j)
        { x = i;
        y = j;
        }
        void prn(void)
        {cout << "x=" << x << "\t";
        cout << "y=" << y << "\n";
        }
        };
        int main()
        {cirscr()
        X O1, O2;
        O1.init();
        O2.init();
        O1.getval(25,12.71);
        O1.prn();
        O2.prn();
        return 0;
        }
```

Sol. The output of the above program will be :
x = 0 y = 0
x = 25 y = 12.71

34. Predict the output of the following program:

```
include<iostream.h>
class A
{
protected :
int a;
public :
A(int x = 1)
{a = x ;
cout << "I am A\n" ; }
~A()
{cout << "Bye-bye from A\n"; }
void f ()
{a + =2;
cout << a << endl;
}
};
class B : public A
{       int b;
        public :
        B() (int y = 5) b = y;
        cout << "Me B\n";
        void f ()
        {b + = 10;
cout << b << endl;
}
~B()
{cout << "Bye-bye from B\n";}
};
int main()
{       A obj1;
        B obj2;
        obj1.f();
        obj2.f();
        obj2.A:: f();
}
```

Sol. **Output:**

```
I       am      A
I       am      A
Me      B
3
15
3
Bye – bye       from B
Bye – bye       from A
Bye – bye       from A
```

35. Answer the questions (i) to (iv) base on the following code :

```
class FaceToFace
{
    char CenterCode[10];
```

```
public:
void Input();
void Output();
};
class Online
{
    char website[50];
    public:
        void SiteIn();
        void SiteOut();
        };
        class Training : public FaceToFace, private Online
    {
        long Tcode;
        float charge;
        int period;
    public:
        void Register();
        void Show();
        };
```

(i) Which type of Inheritance is shown in the above example?

(ii) Write names of all the member functions accessible from Show() function of class Training.

(iii) Write name of all the members accessible through an object of class Training.

(iv) Is the function Output() accessible inside the function SiteOut()? Justify your answer.

Sol. (i) Multiple Inheritance

(ii) Register(), Siteln(), Siteout(), Input(), Output()

(iii) Register (), Show(), Input(), Output()

(iv) No, function Output() is not accessible inside the function SiteOut(), because Output() is a member of class FaceToFace and SiteOut() is a member of class Online, and the classes FaceToFace and Online are two independent classes.

36. What is the difference between Macros and inline function ?

Sol. The difference between macros and inline function is that macros are replaced in the preprocessing stage while inline functions are replaced in the compilation stage.

Note: Memory allocation of various types:

char	1 byte
short	1 byte
int	2 bytes
long	4 bytes
float	4 bytes
double	8 bytes
long double	10 bytes

Chapter - 3	Arrays

Address Calculation:

- Address in Row-major order = $B + w[n(I - \ell_r) + (J - \ell_c)]$
- Address in Column-major order=$B + w[(I - \ell_r) + r(J - \ell_c)]$

B : Base address r : no. of rows I : Row of element given
w : Width/element size n : no. of columns J : Column of element given
ℓ_r : lower row ℓ_c : lower column

- If nothing is given, we assume that the array is stored row-wise, i.e., address is calculated using Row-major.

Eg: An array A[40][10] is stored in the memory along the column with each element occupying 4 bytes. Find out the address of the location A[3][6] if the location A[30] [10] is stored at the address 9000.

- R = 40, N = 10, W = 4 I = 3 J = 6 B = ?

$9000 = B + 4 \times [(30 - 0) + 40 \times (10 - 0)]$

$9000 = B + 4 \times [30 + 400]$

$\Rightarrow$ B = 9000 – 1720 $\Rightarrow$ B = 7280

Address of A[3] [6] = $7280 + 4 \times [(3 - 0) + 40 (6 - 0)$ $= 7280 + 4 \times 243 = 8252$

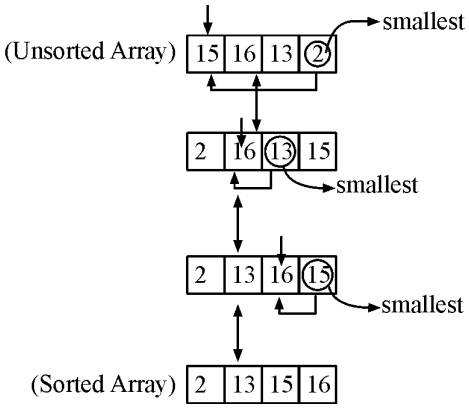
Program in arrays:

(i) Selection sort :

```
void selsort ()
{
  int small, pos, temp ;
  for(int i = 0 ; i < size ; i++)
  {
    small = Ar[i];
    pos = i ;
    for (int j = i+1 ; j < size ; j++)
    {
      if (Ar[j] < small)

    {
      Small = Ar[j];
      pos = j;
      }
      }
      }
      temp = Ar[i];
      Ar[i] = Ar[pos];
      Ar[pos] = temp;
      }
```

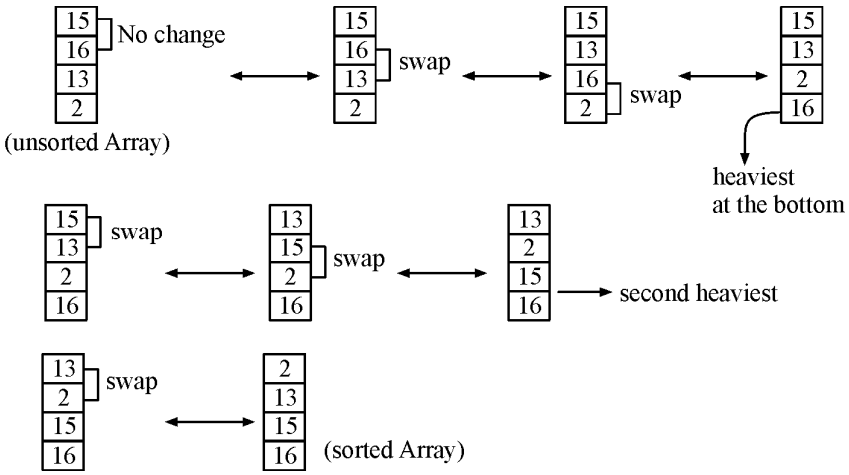
Explanation :



(ii) Bubble sort :

```
void bubsort ()
{
   int temp;
      for (int i = 0; i < size ; i++)
      {
         for(int j = 0; j < (size-1) - i ; j++)
         {
               if (Ar[j] > Ar[j+1])
               {
                     temp = Ar[j] ;
                     Ar[j] = Ar[j+1];
                     Ar[j+1] = temp ;
               }
         }
      }
}
```

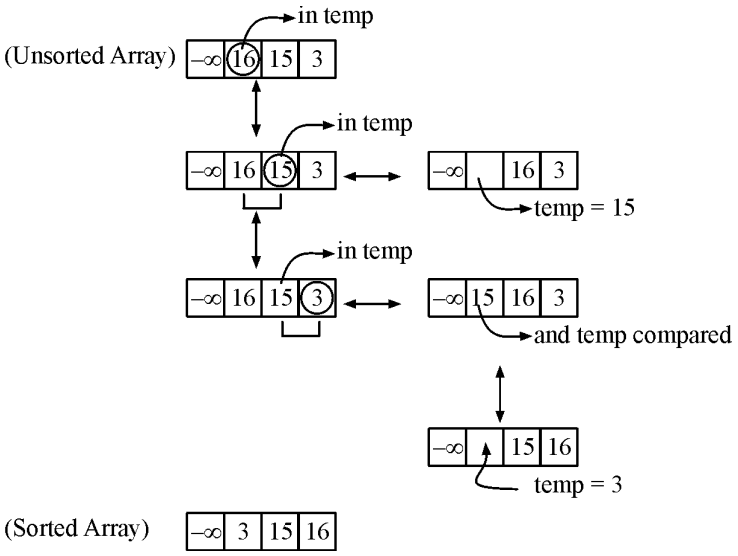
Explanation :



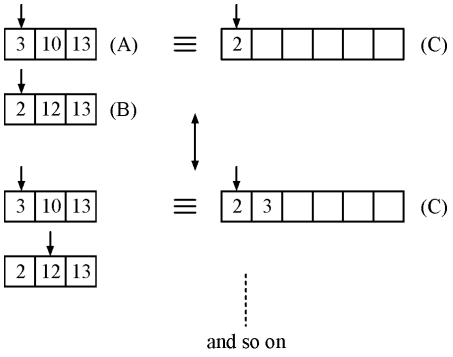
(iii) Insertion sort :

```
void inssort()
{
    int temp, j;
    Ar[0]= int_min;    (use #include <limits.h>)
    for(int i=1; i<=size; i++)
    {
        temp=Ar[i];
        j=i-1;
        while(temp<Ar[j])
        {
            Ar[j+1]= Ar[j];
            j--;
        }
        Ar[j+1]= temp;
    }
}
```

Explanation :



(iv) Merge sort:



1. If two arrays A,B are in ascending order then merge them to form C which is also in ascending order

Sol.

```
void mergesort()
{
   int a, b, c;
   for(a=0, b=0, c=0; a<M&&b<N;)          //M: size of array, N: size of array
   {
        if (A[a]<=B[b])
        C[c++]= A[a++];
        else
        C[c++]= B[b++];
   }
   if (a<M)
   {
       while(a<N → N)
       C[c++]=A[a++];
   }
   else
   {
      while(b<N)
      C[c++]= B[b++];
   }
}
```

2. If an array A is in ascending order and an array B is in descending order, then merge them to form C which is in ascending order.

Sol.

```
void mergesort()
{
   int a, b, c;
   for(a=0. b=N-1, c=0; a<M&&b>=0;)       //M: size of array, N: size of array
   {
        if(A[a]<=B[b])
        C[c++]= A[a++];
        else
        C[c++]= B[b--];
   }
   if(a<M)
   {
       while(a<M)
       C[c++]=A[a++];
   }
```

```
    else
    {
        while(b>=0)
        C[c++]=B[b--];
    }
}
```

(v) Search :

1. Linear search :

```
void linear(int p[],int r)
{
for(int i=0;i<10P-length;i++)
                    if(p[i]==r)
                     {
                    cout<<"\nElement Found!!";
                    cout<<"\nIndex number :"<<i<<" position :"<<(i+1);
                    return; }

cout<<"\nNothing Found!!";
}
int main()
{
system("cls");
int choice,a[10],value,index;
char choices;
cout<<"Enter the elements(10) for the array: ";
for(int i=0;i<10;i++)
cin>>a[i];
do
 {     system("cls");
          cout<<"\nEntert the element to be searched: ";
          cin>>value;
          system("cls");
          linear(a,value);
}
```

2. Binary search :

```
void bubblesort(int r[])
{
int i,j,swapped=1,temp;
for(i=0;((i<9)&&(swapped));i++)
        { swapped=0;
          for(j=9;j>i;j--)
                if(r[j]<r[j-1])
                { swapped=1;
                  temp=r[j];
              r[j]=r[j-1];
                  r[j-1]=temp;
                }
        }
 cout<<"\nYour Array was sorted in Ascending order to perform binary search!!\n";
 for(int k=0;k<10;k++)
        cout<<r[k]<<" ";
 cout<<endl;
}
void binary(int p[],int r)
  bubblesort(p);
  int b=0,l=9,mid;
  while(b<=l)
  {
    mid=(b+l)/2;
    if(p[mid]==r)
    {
cout<<"\nsearch successful. . . . . no. FOUND!!!";
        cout<<"\nIndex: "<<mid<<"Position :"<<(mid+1);
         return;
}
    else if(r>p[mid])
        b=mid+1;

    else
        l=mid-1;
  }
 cout<<"\nnumber Not Found!!";
}
```

```
int main()
{
system("cls");
int choice,a[10],value,index;
char choices;
cout<<"Enter the elements(10) for the array: ";
for(int i=0;i<10;i++)
cin>>a[i];
do
 {     system("cls");
            cout<<"\nEntert the element to be searched: ";
            cin>>value;
            system("cls");
            binary(a,value);    }
```

Chapter - 4	Linked list, Stacks, Queues

Programs in Stacks and Queues:

Stack as an array:

```
#define max 10

class stack
{
int a[max],top;
public:
 stack()
 {top=-1;}
 void push();
 void pop();
 void display();
};

void stack::push()
{
 cout<<" Enter the value: ";
 int val;
 cin>>val;
 if(top==(max-1))
          cout<<" Stack is full !! ";
 else
          a[++top]=val;
}


void stack::pop()
{
 if(top==-1)
          cout<<" Stack is empty !!! ";
```

```
 else
        --top;
}

void stack::display()
{
 for(int i=top;i>=0;--i)
        cout<<a[i]<<endl;
}


int main()
{
int c;
char ch;
stack s;
do{
        system("cls");
        cout<<"Enter your choice\n";
        cout<<"1.push \n";
        cout<<"2.pop \n";
        cout<<"3.Display \n";
        cout<<"4.Exit \n";
        cin>>c;
        switch(c)
                {
                case 1:s.push(); break;
                case 2:s.pop(); break;
                case 3:s.display(); break;
                case 4:exit(0); break;
                }
         cout<<endl<<"Try again?? (y/n)...."<<endl;
         cin>>ch;
 }while(ch=='y'|| ch=='Y');
}
```

Stack as a linked list:

```
struct node{
int value;
node *next;
};

class stack{
node *top;
public:
stack()
{top=NULL;}
void push();
void pop();
void display();
};

void stack::push()
{
```

```
node *ptr;
ptr=new node;
if (ptr==NULL)
        {
        cout<<" memory overflow!!!";
        getch();
        exit(0);
                }
ptr->next=NULL;
cout<<"Enter a number: ";
cin>>ptr->value;
if (top==NULL)
                top=ptr;
else
{
ptr->next=top;
        top=ptr;
}
}

void stack::pop()
{
node *ptr=top;
if (top==NULL)
        {cout<<"Stack is empty!!! ";
        return;}

top=ptr->next;
delete ptr;
cout<<" POPPED OUT OF THE STACK !!";
}
void stack::display()
{
        node *ptr=top;
        if (ptr==NULL)
        {       cout<<"STACK is EMPTY !!";
                return;
                }
        while(ptr!=NULL)
        {       cout<<ptr->value<<" ";
                cout<<endl;
                ptr=ptr->next;
                }
}

int main()
{
char ch;
int choice;
stack s;
do
        {
        system("cls");
        cout<<"What do you wish to do??\n1.push \n";
        cout<<"2.pop \n";
        cout<<"3.Display \n";
```

```
        cout<<"4.Exit\nEnter your choice:";
        cin>>choice;
        switch(choice)
                {
                case 1:s.push(); break;
                case 2:s.pop(); break;
                case 3:s.display(); break;
                case 4:exit(0);
                }
        cout<<"\ntry again ? (y/n) ";
        cin>>ch;
        }while((ch=='y')||(ch=='Y'));
}
```

Queue as an array:

```
#define max 10

class Queue
{
int a[max],front,rear;

public:
 Queue()
 {front=rear=-1;}
 void insertQ();
 void deleteQ();
 void displayQ();
};


void Queue::insertQ()
{
 cout<<" Enter the value: ";
 int val;
 cin>>val;
 if(rear==(max-1))
        cout<<" Queue is full !!";
 else
 a[++rear]=val;
 if(front==rear)
        front=0;
}


void Queue::deleteQ()
{
 if(rear<0)
        cout<<" Queue is empty !!";
 else
        if(front==rear)
        front=rear=-1;
        else
        front++;
 }
```

```
void Queue::displayQ()
{
 for(int i=front+1;i<=rear;++i)
        cout<<a[i]<<" ";
}
int main()
{
int choice;
char choices;
Queue a;
do{
        system("cls");
        cout<<"Enter your choice\n";
        cout<<"1.insert \n";
        cout<<"2.delete \n";
        cout<<"3.Display \n";
        cout<<"4.Exit \n";
        cin>>choice;
        switch(choice)
                {
                case 1:a.insertQ(); break;
                case 2:a.deleteQ(); break;
                case 3:a.displayQ(); break;
                case 4:exit(0); break;
                }
         cout<<endl<<"try again<y/n>"<<endl;
         cin>>choices;
 }while(choices!='n');
}
```

Queues as a linked list:

```
struct node
{
 int value;
 node *next;
};

class Queue
{
        node *front, *rear;
public:
        Queue()
        {front=rear=NULL;}

        void push();
        void pop();
        void display();
};

void Queue::push()
{
  node *ptr;
  ptr=new node;
  if (ptr==NULL)
    {
     cout<<" underflow...!!!!";
```

```
   getch();
   exit(0);
   }
 ptr->next=NULL;
 cout<<"Enter value: ";
 cin>>ptr->value;
 if (front==NULL)
        front=rear=ptr;
 else
  {rear->next=ptr;
   rear=ptr;}
}
void Queue::pop()
{
        node *ptr;
        if (front==NULL)
          {cout<<"Queue is empty ";
                return;}
        front=front->next;
        delete ptr;
        cout<<" Deleted ";
}
void Queue::display()
{
        node *ptr=front;
        if (front==NULL)
        {
                cout<<"queue is empty";
                return;
        }
        while(ptr!=NULL)
        {
                cout<<ptr->value<<" ";
                ptr=ptr->next;
         }
}
int main()
{
char ch;
int choice;
Queue q;
do
{
        system("cls");
        cout<<"What do you wish to do??\n1.push \n";
        cout<<"2.pop \n";
        cout<<"3.Display \n";
        cout<<"4.Exit\nEnter your choice:";
        cin>>choice;
        switch(choice)
        {
                case 1:q.push(); break;
                case 2:q.pop(); break;
                case 3:q.display(); break;
                case 4:exit(0);
        }
```

```
        cout<<"\n try again? (y/n) ";
        cin>>ch;
  }while((ch=='y')||(ch=='Y'));
  }
```

Circular Queue:

```
const int size = 10;
int Q(size] ;
void add (int Q [ ], int front, int &rear, int num)
{       if ((rear + 1) % size = = front)
                cout << "Queue is full" ;
        else
        {
                rear = (rear + 1) % size ;
                Q [rear] = num ;
        }
}
int del (int Q[ ], int & front, int rear)
{       if (front = = rear)
        {
                cout << "Empty Queue" ;
                return (–1) ;
        }
        else
        {       front ++ ;
                int num = Q[front] ;
                return num ;
        }
}
void show (int Q [ ], int front, int rear)
{
        int i = front ;
        if (front = = rear)
        cout << "Empty Queue" ;
        else
        {
        cout << "The queue contains :" ;
                while (i! = rear)
                {
                        i = (i + 1) % size ;
                        cout << Q[i] << 'lt' ;
                }
         }
 }
```

Postfix and Prefix Applications:

1. Convert X ; A + (B* C – (D/E↑F)*G)* H into postfix form showing stack status after every step in tabular form.

Sol.

Symbol Scanned	Stack	Expression Y
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. F	(+ (– (/ ↑	A B C * D E F
14.)	(+ (–	A B C * D E F ↑ /
15. *	(+ (– *	A B C * D E F ↑/
16. G	(+ (– *	A B C * D E F ↑/G
17.)	(+	A B C * D E F ↑ / G * –
18. *	(+ *	A B C * D E F ↑ / G * –
19. H	(+ *	A B C * D E F ↑ / G * – H
20)		A B C * D E F ↑ / G * – H * +

2. Develop the algorithm for evaluating an arithmetic expression A which is in postfix notation. Show the contents of the stack during the execution the execution of the algorithm using the following.

A = 30, 5, 2 ^ 12, 6, / , + –3.

Sol. For algorithm, refer to Algorithm on pate 540 push '(' to the stack and add')' at the end of expression i.e., the expression becomes 30, 5, 2, ^, 12, 6, /, +, – 3)

Step	Input Elemnt/symbol	Action taken	Stack Status	Output
1			(	
2.	30	Push	(, 30	
3.	5	Push	(, 30, 5	
4.	2	Push	(, 30, 5, 2	
5.	^	POP (2 elements)	(, 30	5 ^ 2 = 25
6.		Push the result (25)	(, 30, 25	
7.	12	Push	(, 30, 25, 12	
8.	6	Push	(, 30, 25, 12,6	
9.	/	POP(2 elements)	(, 30, 25	12/6 = 2
10.		Push result (2)	(, 30, 25, 2	
11.	+	POP (2 elements)	(, 30	25 + 2 = 27
12.		Push the result (27)	(, 30, 27	
13.	–	POP (2 elements)	(	30 – 27 = 3
14.		Push the result (3)	(3	
15.	)	POP everything	#	

3. Convert the expression (TURE && FALSE) || (FALSE || TRUE) to postfix expression. Show the content of the stack at every step.

Sol. (TRUE && FALSE) || [FALSE || TRUE)]
Adding] the end of expression and inserting [to the beginning of stack.
Scanning from Left to Right

S.No	Symbol	Stack	Postfix Expression Y
0.		[	
1.	(	[(	
2.	TRUE	_	TRUE
3.	&&	[(&&	__
4.	FALSE	__	TRUE FALSE
5.	)	[	TRUE FALSE &&
6.	||	[||	__
7.	|	[|| !	__
8.	(	[| | !(	__
9.	FALSE	__	TRUE FALSE && FALSE
10.	||	[||!(||	__
11.	TRUE	__	TRUE FALSE && FALSE TURE
12.	)	[|| !	TRUE FALSE && FALSE TRUE ||
13.	]	End of Expression	TRUE FALSE && FALSE TRUE [||]

4. Evaluate the following postfix notation of expression : 50, 60, +, 20, 10, –,*

Sol.

Step	Input symbol	Action taken	Stack Status	Output
1.	50	Push	50 (↑ – top) ↑	
2.	60	Push	50, 60 ↑	
3.	+	Pop (2 elements) Push result (110)	↑ (empty stack) 110 ↑	50 + 60 = 110
4.	20	Push	110, 20 ↑	
5.	10	Push	110, 20 10 ↑	
6.	–	Pop (2 elements) Push result (10)	110 ↑ 110, 10 ↑	20 – 10 = 10
7.	*	Pop (2 elements) Push result (1100)	↑ (empty stack) 1100 ↑	110 * 10 = 1100
8.	End of expression	Pop (result)	Ans = 1100	

5. Evaluate the following postfix notation expression : (show status of Stack after each operation)
False, True, NOT, OR, True, False, AND, OR

Sol.

Step	Input symbol	Action taken	Stack Status	Output
1.	False	Push	False	
2.	True	Push	False, True	
3.	NOT	POP (1 element) Push result (False)	False, False	NOT True = False
4.	OR	POP (2 elements) Push result (False)	False	False OR False = False
5.	True	Push	False, True	
6.	False	Push	False, True, False	
7.	AND	Pop (2 elements) Push result (False)	False, False	True AND False = False
8.	OR	Pop (2 elements) Push result (False)	False	False OR False = False Ans. = False

Theory Questions (Arrays, Linked lists, Stacks and Queues)

1. Define a pointer.

Sol. A pointer is a memory variable that can store memory address.

2. If arr is an array of integers, why is the expression arr++ not legal?

Sol. Because name of the array is the constant pointer (which stored base address of the first elements) and arr++ would attempt to change the value of the constant.

3. How is &p different from*p?

Sol. &p gives us the address of variable p and *p dereferences p and gives us the value of stored in memory location pointed to by p.

4. Define the term data structure.

Sol. A data structure is a named group of data of different data types which can be processed as a single unit.

5. What is meant by the term 'Overflow' and 'Underflow" ?

Sol. 'Overflow' means attempt to INSERT when the list is full and no free space is available.
'Underflow' means attempt to DELETE an item from an empty list.

6. How does C++ organize memory when a program is run?

Sol. Once a program is compiled, C++ creates four logically distinct regions of memory :

(i) area to hold the compiled program code
(ii) area to hold global variables
(iii) the stack area to hold the return addresses of function calls, arguments passed to the functions, local variables for functions, and the current state of the CPU.
(iv) the heap are from which the memory is dynamically allocated to the program.

7. Differentiable between static and dynamic allocation of memory.

Sol. In the static memory allocation, the amount of memory to be allocated is predicted and preknown. This memory is allocated during the compilation itself. All the declared variables declared normally, are allocated memory statically.

In the dynamic memory allocation the amount of memory to be allocated is not known beforehand. It is allocated during run-time as and when required. The memory is dynamically allocated using new operator, the memory allocation operator C++.

The objects that are allocated memory statically have the lifetime as their scope allows, as decided by the compiler. And the objects that are allocated memory dynamically have the lifetime as decided by the programmer. That is, until the programmer explicitly deallocated the memory, such object live in the memory.

8. What do you understand by memory leaks? What are the possible reasons for it? How can memory leaks be avoided?

Sol. If the objects, that are allocated memory dynamically, are not deleted using **delete**, the memory block remains occupied even at the end of the program. Such memory blocks are known as orphanded memory blocks.
These orphaned memory blocks when increase in number, bring as adverse effect on the system. This situation is known as memory leak. The possible reasons for this are:

(i) a dynamically allocated object not deleted using **delete.**
(ii) delete statement is not getting executed because of some logical error.
(iii) assigning the result of a **new** statement to an already occupied pointer.

The memory leaks can be avoided by

(i) making sure that a dynamically allocated object is deleted.
(ii) a **new** statement stores its return value (a pointer) in a fresh pointer.

9. What is 'this' pointer ? What is its significance?

Sol. The member functions of every object have access to a sort of magic pointer named **this**, which points to the object itself. Thus any member function can find out the address of the object of which it is a member.
The this pointer represents an object that invokes a member function. It stores the address of the object that is invoking a member function and it (**this** pointer) is an implicit argument to the member function being invoked
The this pointer is useful in returning the object (address) of which the function is a member.

10. Distinguish between
int *ptr = new int(5);
int *ptr = new int[5];

Sol. The first statement allocates memory of one integer to **ptr** and initializes it with value 5. The second statement allocates memory of 5 continguous integers (i. e., an array of 5 integers) and stores beginning address in pointer **ptr.**

11. In which two ways can the elements of a double dimensional array may be stored in computer's memory?
Or
How is computer memory allotted for a two – dimensional array?

Sol. : For two – dimensional array, the computer memory is allocated either in Row-major form or Column major form.
Row Major form stores the 2-D array row wise i.e, firstly the first row is stored, then the second row, then third row, and so forth.
Column Major form stores the 2-D array column wise i.e., firstly the first column, then the second column, then third, and so forth. The default form is Row-major.

12. Write a function SKIPEACH (int H[][3], int C, intR) in C++ to display all alternate elements from two – dimensional array H (starting from H[0][0]).
For Example :
If the array is containing :
12 45 67
33 90 76
21 43 59
Total output will be : 12 67 90 21 59

Sol.
```
viod SKIPEACH(int h[][3], intC, intR)
 {
   int display = 1;
   for (int i = 0 ; i < R ; i++)
   for (int j = 0; j < C ; j++)
   {
      if(display = = 1)
      count << h[i][j];
      display * =  1;
            }
   }
```

13. Distinguish between infix, prefix and postfix algebraic expressions given examples of each.

Sol. Infix Notation. In this notation, the operator symbol is placed in between the operands e.g.,
A + B , (A – C) × B
Prefix Notation. In this notation, the operator symbol is placed before its operands e.g.,
+ AB × –ACB
Postfix Notation. In this notation, the operator symbol is placed after its operands e. g.,
AB+, AC – B×, ABC×+,

14. An algorithm requires two stacks of sizes M and N that are to be maintained in the memory. Illustrate with an example how will you adjust two stacks in one dimensional array with M + N memory locations so that the overflow condition in minimized.

Sol. Let us say that stack A is with size M and stack B is with size N.

If the stack A is stored in locations 0 to M – 1 and the stack B is stored in locations M to M + N – 1, the separate areas for the two are ensured. In the beginning the TOPs of both stacks are at opposite ends. When TOP. A reaches at M –1, any further insertion will lead to overflow in stack A and when TOP.B reaches at M, any further insertion in stack B will lead to overflow.

Chapter - 5	Data File Handling

Theory Questions (Data File Handling)

1. What is the difference between functioning of ios : : ate and ios : : app file modes?

Sol. Both ios :: ate and ios :: app place the file pointer at the end of the file just opened. The difference between the two is that ios :: app lets you add data to the end of the file only, while the ios :: ate mode allows you to write data anywhere in the file, even over old data.

2. What is difference between get() and read() ?

Sol. The read() declared under iostream.h extracts a given number of characters into an array.
The get() declared under isotream.h either extracts the next character or EOF or extracts characters into a char * until eof or delimiter encountered or specified (len-1) bytes have been read.

3. What is the difference between put() and write()?

Sol. The prototypes of put() and write() are :
ostream & put(char ch);
ostream & write((char *) & buf, int sizeof(buf));
The put( ) writes the value of ch (one character) to the stream and returns a reference to the stream. The write( ) function writes sizeof(buf) bytes to the associated stream from the buffer pointed to by buf. The data written to a file using write( ) can only be read accurately using read( ).

4. Distinguish between Serial files and Sequential files.

Sol. In a serial file records are arranged in the way they are inserted in a file. There is no significance of primary key in a serial file.
In a sequential file, records can be arranged in a particular order depending upon primary key value of the records. Records are accessed in the same order as they are entered in. However, same data can be stored in different files in different order of different keys.

5. Difference between getline() and getc() functions.

Sol. getc( ) function can read one character at a time. On the other hand, getline() can read a line of text of specified size. getc() is defined in stdio.h however getline() is defined in isotream.h.

6. Observe the program segment given below carefully and answer the questions that follow :

```
#include<fstream.h>
class Book
{       int Bno ; char Title [20]
        public ;
        void EnterVal( )
{       cin >> Bno : cin.getline(title, 20); }
        void ShowVal( )
        { cout << Bno << "#" << Title <<endl ;}
};
void Search(int RecNo)
{
        fstream File ;
        Book B;
        File.open ("BOOK.DAT", ios::binary|ios::in);
        _____          //Statement 1
```

```
        File.read((char*)&B, sizeof(B)) ;
        B.ShowVal( );
        File.close();
}
void Modify(int RecNo)
{
        fstream File;
        Book B;
        File.open ("BOOK.DAT", ios::binary|ios::in|ios::out) ;
        B.EnterVal( );
        _____          //Statement 2
        File.write((char*)&B, sizeof(B));
        File.close( );
}
```

(i) Write statement 1 to position the file pointer to the beginning of the desired record to be read, which is sent as parameter of the function (assuming RecNo 1 stands for the first record)

(ii) Write statement 2 to position the file pointer to the beginning of the desired record to be modified, which is sent as parameter of the function (assuming RecNo 1 stands for the first record)

Sol. (i) File.seekg((RecNo – 1) *sizeof(B)) ; //Statement 1

(ii) File.seekp((RecNo – 1)*sizeof(B)); //Statement 2

7. How are binary files different from text files in C++ ?

Ans. When a file is opened in text mode, various character translations may take place, such as the conversion of carriage-return and linefeed sequences into newlines. However, no such character translations occur in files opened in binary mode. Any file can be opened in either text or binary mode. The only difference is the occurrence of character translations in text mode.

8. What is the need and usage of read( ) and write( ) functions when there are get( ) and put( ) functions for I/O?

Sol. The get( ) and put( ) functions perform I/O byte by byte. On the other hand, read( ) and write( ) functions let you read and write structures and objects in one go without creating the need for I/O for individual constituent fields.

9. What is the difference between get( ) and getline( ) functions even when they share similar prototypes?

Sol. The difference between get(buf, num, delim) and getline(buf, num, delim) is that even though both read characters from the input stream into a character array being pointed to buf until either num number of character are read or the character specified by delim (by default '/n') is encountered, whichever occurs earlier. The major difference is that get( ) does not extract the delimiter character from the input stream *i.e.*, the delimiter character remains in the stream after get( ) is over. On the other hand getline( ) does extract the delimiter newline character from the input stream so that the stream is empty after getline( ) is over.

10. What is a stream ? Name the streams generally used for file I/O.

Sol. A stream is a sequence of bytes. Or in other words, a stream is a flow of bytes into or out of a program. Generally three streams are used for file I/O. These are :

(i) ifstream. It is derived from istream class and it is used to associate a file with an input buffer so that the file can be read from.

(ii) ofstream. It is derived from ostream class and it is used to associate a file with an output buffer so that the file can be written onto.

(iii) fstream. It is derived from iostream class and it is used to associate a file with a buffer so that the file can read from as well as written onto.

11. What are file modes? What role do they play in file I/O ?

Sol. The filemode describes how a file is to be used in the program. That is whether the file is to be read from only, to be written to it or to be appended, and so on. The filemode(s) are specified at the time of opening files. Different filemode constants defined in ios class are :

ios : : in	opens file for reading
ios : : out	opens file for writing
ios : : ate	seeks to eof upon opening file
ios : : app	appends to end of file
ios : : trunk	truncates file if it exists
ios : : nocreate	causes open fail if file does not exist
ios : : noreplace	causes open fail if file does exist
ios : : binary	opens file in binary mode.

Programs in Data file Handling :

1. Program to search in a file having records maintained through classes.

```
#include<iostream.h>
#inclued<fstream.h>
class stu  {        int rollno ;
                    char name[25] ;
                    char Class[4] ;
                    float marks ;
                    char grade ;
public:
        void getdata( )
{       cout << "Rollno:" ;          cin >> rollno ;
        cout << "Name:" ;            cin >>  name ;
        cout << "Class:" ;           cin >> Class ;
        cout << "Marks:" ;           cin >> marks ;
        if (marks >= 75)  grade = 'A' ;
        else if(marks >= 60)        grade = 'B' ;
        else if(marks >= 50)        grade = 'C' ;
        else if(marks >= 40)        grade = 'D' ;
        else grade = 'F' ;
}
void putdata( )
{       cout << name << ", rollno"<< rollno<<"has"<< marks
                << "% marks and" << grade <<"grade."<< endl ;
}
int getrno( )  //accessor function
{       return rollno ;
}
}s1 ;
int main( )
{
```

```
        int rn ;
        char found = 'n' ;
        ifstream fi("stu.dat",ios::in);          //stu.dat must exist on disk
                cout <<" Enter rollno to be searched for:" ;
                cin >> rn ;
        while(!fi.eof( ))
        {       fi.read((char*)&s1,sizeof(s1));
                if (s1.getrno( ) == rn)
                {
                        s1.putdata( );
                        found = 'y' ;
                        break ;
                }
        }
        if (found == 'n')
            cout << "Rollno not found in file!!" << endl ;
        fi.close( ) ;
        return  0;
}
```

Output:

Enter rollno to be searched for : 108
Venkat, rollno 108 has 78 % marks and A grade.
Enter rollno to be searched for : 119
Rollno not found in file !!

2. Program to append data in a file.

```
#include<iostream.h>
#inclued<fstream.h>
class stu {     int rollno ;
                char name[25] ;
                char Class[4] ;
                float marks ;
                char grade ;
            public:
                void getdata( )
                {       cout << "Rollno:" ;      cin >> rollno ;
                        cout << "Name:" ;        cin >> name ;
                        cout << "Class:" ;       cin >> Class ;
                        cout << "Marks:" ;       cin >> marks ;
                        if (marks >= 75) grade = 'A' ;
                        else if(marks >= 60)     grade = 'B' ;
                        else if(marks >= 50)     grade = 'C' ;
                        else if(marks >= 40)     grade = 'D' ;
                        else grade = 'F' ;
                }
                void putdata( )
```

```
        {       cout << name <<" , rollno" << rollno << "has" << marks
                        << "%marks and" <<      grade <<" grade." << endl;
        }
        int getrno( )  //accessor function
        {       return rollno ;
        }
} s1 ;
        int main( )
        {       ofstream fo("stu.dat",ios::app) ;
                char ans = 'y' ;
                while (ans = = 'y')
        {s1.getdata( );
         fo.write((char*)&s1, sizeof(s1)) ;
        cout << "Record added to file.\n" ;
        cout << "Want to enter more records? (y/n)…" ;
        cin >> ans ;
        }
        fo.close( );
        return 0 ;
}
```

Output:

Rollno : 109
Name : Tauqueer
Class : XIA
Marks : 79
Record added to file.
Want to enter more records?(y/n)..n

3. Program to insert data in a sorted file.

```
#include<iostream.h>
#include<fstream.h>
#include<stdio.h>                       //for rename( ) and remove ( )
class stu {     int rollno ;
                char name[25]
                char Class[4];
                float marks ;
                char grade ;
        public:
                void getdata( )
                {       cout << "Rollno:" ;      cin >> rollno ;
                        cout << "Name:" ;       cin >> name ;
                        cout << "Class:" ;      cin >> Class ;
                        cout << "Marks:" ;      cin >> marks ;
                        if (marks > = 75) grade = 'A' ;
                        else if(marks > = 60)   grade = 'B' ;
                        else if(marks > = 50)   grade = 'C' ;
                        else if(marks > = 40)   grade = 'D' ;
```

```
                else grade = 'F' ;
        }
                void putdata( )
                {       cout << "Rollno" << rollno << "\tName:" << name
                        << "\n Marks: " << marks << "\t Grade: " << grade<< endl;
                }
                int getrno( )  //accessor function
                {       return rollno ;         }
        }s1, stud :
int main( )
{       ifstream fi("stu.dat", ios::in) ;        //stu.dat must exist on disk
        ofstream fo("temp.dat",ios::out) ;
        char last = 'y' ;
        cout << "Enter details of student whose record is to be inserted\n " ;
        s1.getdata( ) ;
        while(!fi.eof( ) )
        {       fi.read((char*)&stu, sizeof(stud));  //s1 is new record, stud is record from file
                if (s1.getrno() <= stud.getrno( ) )
                {       fo.write((char*)&s1, sizeof((s1)) ;
                        last = 'n' ;
                        break ;
                }
                else
                        fo.write((char*)&stud, sizeof(stud)) ;
        }
        if (last = = 'y')
                fo.write((char*)&s1, sizeof(s1)) ;
        else if (!fi.eof( ) )
{       while(!fi.eof( ) )
        {       fi.read((char*)&stud, sizeof(stud)) ;
                fo.write((char*)&stud, sizeof(stud));
        }
}
fi.close( );
fo.close( );
remove("stu.dat") ;
rename("temp.dat", "stu.dat" ) ;
fi.open("stu.dat", ios::in)
cout < < "File now contains\n" ;
while(!fi.eof( ) )
{       fi.read ( (char*)&stud, sizeof(stud)) ;
        if (fi.eof( )) break ;
        stud.putdata( );
}
fi.close( );
return 0 ;
}
```

Output:

Enter the details of student whose record is to the entered :

Rollno : 115
Name : Neha
Class : XIA
Marks : 88

File now contains :

Rollno 102 Name : Joseph
Marks : 67 Grade : B
Rollno 104 Name : Simran
Marks : 77 Grade : A
Rollno 105 Name : Manya
Marks : 49 Grade : D
Rollno 107 Name : Tilotma
Marks : 65 Grade : B
Rollno 109 Name : Tauqueer
Marks : 79 Grade : A
Rollno 115 Name : Neha
Marks : 88 Grade : A

4. Program to delete a record form a file.

```
#include<iostream.h>
#include<fstream.h>
#include<stdio.h>                          //for rename( ) and remove ( )
#include<string.h>
class stu {        int rollno ;
                   char name[25]
                   char Class[4];
                   float marks ;
                   char grade ;
           public:
                   void getdata( )
                   {       cout << "Rollno:" ;        cin >> rollno ;
                           cout << "Name:" ;          cin >> name ;
                           cout << "Class:" ;         cin >> Class ;
                           cout << "Marks:" ;         cin >> marks ;
                           if (marks >= 75)  grade = 'A' ;
                           else if(marks >= 60)       grade = 'B' ;
                           else if(marks >= 50)       grade = 'C' ;
                           else if(marks >= 40)       grade = 'D' ;
                           else grade = 'F' ;
                   }
                   void putdata( )
                   {       cout << "Rollno" << rollno << "\tName :" << name
                                   << "\n Marks : " << marks << "\tGrade:" << "grade." << endl;
                   }
                           int getrno( )  //accessor function
```

```
                    {       return rollno ;         }
            }s1, stud ;
int main( )
{           ifstream fio("stu.dat", ios::in) ;          //stu.dat must exist on disk
            ofstream file("temp.dat" , ios::out) ;
            int rno ;
            char found = 'f', confirm = 'n' ;
            cout << "Enter rollno of student whose record is to be deleted \n " ;
            cin >> rno ;
            while(!fio.eof( ) )
            { fio.read((char*)&s1, sizeof(s1)) ;
              if (s1.getrno ( ) == rno )
              {     s1.putdata( ) ;
                    found = 't' ;
                    cout << " Are you sure, you want to delete this record? (y/n).." ;
                    cin >> confirm ;
                    if (confirm == 'n')
                          file.write( (char*)&s1, sizeof(s1)) ;
            }
            else
               file.write((char*)&s1, sizeof(s1)) ;
            }
            if(found == 'f')
            cout << "Record not found !!\n" ;
            fio.close( ) ;
            file.close( ) ;
            remove(" stu.dat") ;
            rename("temp.dat", "stu.dat");
            fio.open("stu.dat", ios: :in);
            cout << "Now the file contains\n";
            while(!fio.eof( ) )
            {
                    fio.read((char*)&stud, sizeof(stud)) ;
                    if (fio.eof( ) ) break ;
                    stud.putdata( );
            }
            fio.close( ) ;
            return 0 ;
}
```

Output:

Enter rollno of student whose record is to be deleted
107
Rollno 107 Name : Tilotma
Marks : 65 Grade : B
Are you sure, you want to delete this record? (y/n)..y
Now that file contains
Rollno 101 Name : Abhinav

```
Marks : 89          Grade : A
Rollno  102         Name : Joseph
Marks : 67          Grade : B
Rollno  103         Name : Naureen
Marks : 59          Grade : C
Rollno  104         Name : Simran
Marks : 77          Grade : A
Rollno  105         Name : Manya
Marks : 49          Grade : D
Rollno  106         Name : David
Marks : 88          Grade : A
Rollno  109         Name : Tauqueer
Marks : 79          Grade : A
Rollno  115         Name : Neha
Marks : 88          Grade : A
```

5. Program to modify data in a given file.

```
#include<iostream.h>
#include<fstream.h>
#include<stdlib.h>                      //for rename( ) and remove ( )
#include<stdio.h>
#include<string.h>

class stu {         int rollno ;
                    char name[25]
                    char Class[4];
                    float marks ;
                    char grade ;
          public:
                    void getdata( )
                    {       cout << "Rollno:" ;      cin >> rollno ;
                            cout << "Name:" ;       cin >> name ;
                            cout << "Class:" ;      cin >> Class ;
                            cout << "Marks:" ;      cin >> marks ;
                            if (marks >= 75)  grade = 'A' ;
                            else if(marks >= 60)    grade = 'B' ;
                            else if(marks >= 50)    grade = 'C' ;
                            else if(marks >= 40)    grade = 'D' ;
                            else grade = 'F' ;
                    }
                    void putdata( )
                    {       cout << "Rollno" << rollno << "\tName :" << name
                                    << "\n Marks : " << marks << "\tGrade:" << grade << endl;
                    }
                    int getrno( )                   //accesor function
                    {       retrun rollno ;     }
                    void modify( ) ;
```

```
        } s1, stud :
        void  stu::modify( )
        {       cout << "Rollno : " < < rollno < < endl ;
                cout << "Name : " << name << "\t Class : " << Class
                     << "\t Marks : "<< marks << endl ;
                cout << "Enter new details. " < < endl ;
                char nm[20] = " " , C[4] = " " ;
                float mks ;
                cout << "New Name : (Enter     to retain old one)" ;
                cin >> nm ;
                cout << "New Marks : (press  –1 to retain old one)" ;
                cin >> mks ;
                if (strcmp(nm, " . ")! = 0)
                        strcpy(name, nm) ;
                if (strcmp(Cl, " . ") ! = 0)
                        strcpy(Class, Cl) ;
                if (mks ! =  -1 )
                {       marks = mks ;
                        if (marks >= 75)  grade = 'A' ;
                        else if(marks >= 60)        grade = 'B' ;
                        else if(marks >= 50)        grade = 'C' ;
                        else if(marks >= 40)        grade = 'D' ;
                        else grade = 'F' ;
                }
        }
int main( )
{       fstream fio("stu.dat", ios::in|ios::binary);      //stu.dat must exist on disk
        int rno; long pos; char found = 'f' ;
        cout << "Enter rollno of student whose record is to be modified\n " ;
        cin >> rno ;
        while(!fio.eof( ) )
        {       pos = fio.tellg( ) ;
                fio.read((char*)&s1, sizeof(s1)) ;
                if (s1.getrno( ) == rno )
                {       s1.modify( ) ;
                fio.seekg(pos) ;
                fio.write((char*)&s1, sizeof(s1)));
                found = 't'
                break ;
        }
        }
        if(found = ='f')
        cout << "Record not found!!\n" ;
        fio.seekg(0) ;
        cout << 'Now the file contains\n" ;
        while(!fio.eof( ) )
        {       fio.read( (char*)&stud, sizeof(stud)):
```

```
                stud.putdata( ) ;
            }
            fio.close( ) ;
            return 0 ;
        }
```

Output:

```
Enter rollno of student whose record is to be modified
107
Rollno : 107
Name : Tilotma          Class : XIA          Marks : 65
Enter new details.
New Name : (Enter ' . ' to retain old one) .
New Class : (press ' . ' to retain old one) .
New Marks : (press ' -1 ' to retain old one) 72
Now the file contains

Rollno 101              Name : Abhinav
Marks : 89              Grade : A
Rollno   103            Name : Naureen
Marks : 59              Grade : C
Rollno   104            Name : Simran
Marks : 77              Grade : A
Rollno   106            Name : David
Marks : 88              Grade : A
Rollno 107              Name : Tilotma
Marks : 72              Grade : B
Rollno : 108            Name : Venkat
Marks : 78              Grade : A
Rollno   109            Name : Tauqueer
Marks : 79              Grade : A
```

Chapter - 6 & 7	Database and SQL

❒ **Important Points :**

- **Database :** Collection of logical units/ inter-related data such as 'table', 'index' etc.
- **Table :** Collection of Records & Fields.
- **Tuples :** Rows of a relation (table) are known as Tuples.
- **Attributes :** Columns of a Relation are known as Attributes.
- **Degree :** Number of Columns in a Relation.
- **Cardinality :** Number of Rows/Tuples in a Relation.
- All the Databases come under Generation IV Language.

❑ **Keys :**

Key is a Column. Types of Keys are :

(i) **Primary Key :** It is a key which is 'unique' and 'not null'. It is used to set relation among the tables. 'Unique' : Cannot be duplicated 'Not Null' : Cannot be Empty

(ii) **Candidate Key :** Those keys which are 'candidate' to become a Primary Key are 'Candidate keys'.

(iii) **Alternate Key :** It is a Candidate Key which is not a Primary Key.

(iv) **Foreign Key :** It is a key in another table which is having same data-type and size as that of Primary Key in the Main or First table.

✓ The name of the Foreign Key can be different from the Primary Key.

Example :

Table Class II

Admn No.	Roll No.	Name	Marks
1011	1	Rahat	85
1083	2	Irfan	75
2011	3	Maya	63
1000	4	Sham	60
999	5	Zoya	92

In the above table, columns admn. No. and Roll no. have unique value for each row, so both are candidates to become Primary key hence both of these columns are candidate keys. Out of these 2 we can assign one as Primary key and other will become alternate key

Candidate keys : Admn No, Roll Nol
Primary Key : Roll No.
Alternate Key : Admn No.

➢ **SQL : Structured Query Language :**

This language is for all DBMS and RDBMS. It is furthered divided into certain categories:

- DDL : Data Definition Language
- DML : Data Manipulation Language
- DQL : Data Query Language

DDL	DML
➢ Data Definition Language ➢ Works with a table structure ➢ Contains 3 statements : Create, Alter and Drop	➢ Data Manipulation Language ➢ Works with the records of the table ➢ Contains 3 statements : Insert, Update and Delete

❑ DDL Statements

- ➢ Create : To create a table, create statement is used.
- ➢ Alter : It is used to add/modify/delete a column.
- ➢ Drop : To drop or delete the complete table. The table structure as well as the records are completely removed from the memory.

❑ DML Statements

- ➢ Insert : To insert records in the table.
- ➢ Update : To edit or make changes in the existing record.
- ➢ Delete : To delete records from the table.

❑ DQL Statements

Contains One Single statement : 'Select'.

'Select' statement is used to display the records of the table.

❑ Things to Remember

- ➢ Commands are **Case-Retentive** (i.e. not Case Sensitive).
- ➢ Data in the table is **Case-Sensitive**.
- ➢ **Data-types Used :** int, float, double, char, date, varchar.

  Varchar : Variable Character, allocates memory dynamically, unlike 'char'.
- ➢ **Wild Card Characteristics: '_' & '%'**

  '_' : Used for single char of any type

  '%' : Used for multiple char of any type
- ➢ **'Group By' Clause :** It works on lots of records together. It combines a pool of values in a particular field to show certain grouping of data.
- ➢ **Cartesian Product**

  The product of tuples of two or more than two tables. It has no relevance as it multiplies the records of two or more than two tables without any criteria.

  To rectify the problem & to produce relevant Output, 'Joins' are used!

❑ Joins :

- ➢ Used to join two or more than two tables using a particular operator.
- ➢ 'Equi-Join' is a join which combines two or more than two tables using 'equal to (=)' operator.

Theory Questions (Database and SQL)

1. What is relation? Define the relational data model

Sol. **Relational Model.** The relational model represents data and relationship among data by a collection of tables known as relations, each of which has a number of columns with unique names.

2. Give example of functional dependency?

Sol. Given a relation *R*, attribute *Y* of *R* is functionally dependent on attribute *X* of *R* if and only if each *X*-value in *R* has associated with it precisely one *Y*-value in *R* (at any one time).

Let us consider a relation R_x.

R_x

J	*K*	*L*
X	1	0
X	1	6
Y	4	1
Y	4	9
Z	3	5

$J \rightarrow K$ (*K* is functionally dependent on *J*)

$J \rightarrow L$ (L is not functionally dependent on *J*)

In relation R_x *K* is functionally dependent upon *J* since for each value of *J*, there is only one value of *K* associated with. On the other hand *L* is not functionally dependent on *J* as it is not possible to uniquely determine the value of *L* corresponding to a give value of *J*.

3. What is data redundancy? What are the problems associated with it

Sol. Duplication of data is redundancy. It leads to the problems like wastage of space and data inconsistency.

4. What is data inconsistency?

Sol. The problem of multiple mismatched copies of same data, in a database, is called data inconsistency. It is the result of unsupervised data redundancy.

5. Name three levels of data abstraction in a database.

Sol. (i) Internal (Physical) level
(ii) Conceptual (Logical) level
(iii) External (View) level

6. Given a suitable example of a table with sample data and illustrate Primary and Candidate Keys in it.

Sol. **Candidate key** A candidate key is the one that is capable of becoming primary key *i.e.*, a field or attribute that has unique value for each row in relation.

Primary key is a designated attribute or a group of attributes whose value can uniquely identify the tuples in the relation.

Table Class 11

AdmNo	RollNo	Name	Marks
1011	1	Rahat	85
1083	2	Irfan	75
2011	3	Maya	63
1000	4	Shaun	60
999	5	Sukhi	92
1200	6	Zoya	86

In the above table, column AdmNo and RollNo have unique values for each row, so both are candidate to become primary keys. Hence both of these column are Candidate Keys. Out of these two we can assign one as primary key and other will become alternate key.

7. What do you understand by Selection and Projection operations in relational algebra?

Sol. **Selection** means selecting some row (tuples) from according to given condition

e.g., $\sigma_{\text{price}>20.2}(\text{Item})$

Project Operation yields a vertical subset of a given relation (*i.e.,* select all tuples containing only given column of a relation).

e.g., $\pi_{\text{NAME, DESIG}}(\text{Employee})$

8. Explain Cartesian product of two relations.

Sol. The Cartesian product is binary operation and is denoted by a cross (*x*). The Cartesian product of two relation A and B is written as $A \times B$. The Cartesian product yields a new relations which has a degree (Number of attributes) equal to the sum of the degrees of the two relations operated upon. The number of tuples (cardinality) of the new relation is the product of the number of tuples of the two relations operated upon. The Cartesian product of two relations yields a relation with all possible combinations of the tuples of the two relations operated upon.

9. What do you understand by Union and Cartesian Product operations performed upon two relations?

Sol. **Cartesian Product.** The Cartesian product of two relations A and B is written as $A \times B$. The Cartesian product of two relations yields a relation with all possible combinations of the tuples of the two relations operated upon *e.g.,* given two relations Student and Instructor as shown below :

Student

Stud#	Stud-Name	Hosteler
S001	Meenakshi	Y
S002	Radhika	N
S003	Abhinav	N

Instructor

Inst#	Inst-Name	Subject
101	K. Lal	English
102	R.L. Arora	Math

The Cartesian product of these two relations, **Student × Instructor**, will yield a relation as :

Stud#	Stud-Name	Hosteler	Inst#	Inst-Name	Subject
S001	Meenakshi	Y	101	K.Lal	English
S001	Meenakshi	Y	102	R.L. Arora	Maths
S002	Radhika	N	101	K.Lal	English
S002	Abhinav	N	102	R.L.Arora	Maths
S003	Abhinav	N	101	K.Lal	English
S003	Abhinav	N	102	R.L. Arora	Maths

Union. The union operation produces a third relation that contains tuples from both the operand relations which must be union-compatible. To denote the union of two relations X and Y, we write as $X \cup Y$ which will contain all tuples of Y in it.

❑ Syntax of the Statements :

1. Create :

Create table <table name> (<datatype><name of the field><size> . . . . . .) ;

2. Alter :

Alter table <table name> add . . . . . ;

Alter table <table name> modify . . . . . ;

Alter table <table name> drop . . . . . .;

3. Drop :

Drop table <table name> ;

4. Insert :

Insert into <table name> values (. . . . . .) ;

5. Update :

Update <table name> set dob = '2001/11/05' ; //All dob will become 2001/11/05

- 'Where' clause is used to set criteria on records.

Update <table name> set dob = '2001/11/05' where <condition> ;

6. Delete :

Delete from <table name> where <condition> ;

7. Select :

Select * from <table name> where <condition> ;

'*' is used to display all the fields

Logical operators : AND OR NOT

Other Important Operators : IN LIKE

- Some Important Group functions : Which work on number of rows.

sum ( )	:	calculates sum of a particular field, i.e., all the values present in a field.
avg ( )	:	finds out the average of values.
min ( )	:	picks up the minimum value in the given range.
max ( )	:	picks up the maximum value in the given range.
count ( )	:	counts the values in a field.

- 'Order By' clause

For eg.: select name from <table name> order by <field name> asc ;

asc = ascending order desc = descending order

If nothing is mentioned, Order will be in Ascending Order!!

- 'Having' Clause : The 'Having' Clause places conditions on groups in contrast to 'Where' clause that places conditions on individual rows. While 'Where' conditions cannot include aggregate functions, 'Having' conditions can do so.

For eg.: Select avg(gross), sum(gross) from employee Group By grade Having grade = 'E4';

Select job, count(*) from emp Group By job Having count(*)<3;

❑ Solved Problems :

1. Given the following student relation :

Relation Student :

No.	Name	Age	Department	Date of adm	Fee	Sex
1.	Pankaj	24	Computer	10/01/97	120	M
2.	Shalini	21	History	24/03/98	200	F
3.	Sanjay	22	Hindi	12/12/96	300	M
4.	Sudha	25	History	01/07/99	400	F
5.	Rakesh	22	Hindi	05/09/97	250	M
6.	Shakeel	30	History	27/06/98	300	M
7.	Surya	34	Computer	25/02/97	210	M
8.	Shikha	23	Hindi	31/07/97	200	F

Write SQL commands for (a) to (f) and write output for (g).

(a) To show all information about the students of History department.

(b) To list the names of female students who are in Hindi department.

(c) To list names of all students with their date of admission in ascending order.

(d) To display student's Name, Fee, Age for Male Students only.

(e) To count the number of students with Age < 23.

(f) To insert a new row in the 'Student' table with the following data :
9, "Zaheer", 36, "Computer", {12/03/97}, 230, "M"

(g) Give the output of the following SQL statements:

(i) Select COUNT (distinct Department) from Student;

(ii) Select MAX(Age) from Student where Sex = "F";

(iii) Select AVG(Fee) from Student where Dateofadm < {01/01/98};

(iv) Select SUM(Fee) from Student where Dateofadm < {01/01/98};

SOLUTION :

(a) Select * from Student where Department = "History";

(b) Select Name from Student where Sex = "F" and Department = "Hindi";

(c) Select Name from Student order by Dateofadm; (By default ascending order only taken)

(d) Select Name, Fee, Age from Student where Sex = "M";

(e) Select COUNT(*) from Student where Age < 23;

(f) Insert into Student Values (9, "Zaheer", 36, "Computer", {12/03/97}, 230, "M");

(g) **(i)** 3 **(ii)** 25

(iii) 216 **(iv)** 1080

2. Given the following table Hospital:

Table : Hospital

No.	Name	Age	Department	Date of adm	Charges	Sex
1.	Sandip	65	Surgery	23/02/98	300	M
2.	Ravina	24	Orthopedic	20/01/98	200	F
3.	Karan	45	Orthopedic	19/02/98	200	M
4.	Tarun	12	Surgery	01/01/98	300	M
5.	Zubin	36	ENT	12/01/98	250	M
6.	Ketaki	16	ENT	24/02/98	300	F
7.	Ankita	29	Cardiology	20/02/98	800	F
8.	Zareen	45	Gynecology	22/02/98	300	F
9.	Kush	19	Cardiology	13/01/98	800	M
10.	Shailya	31	Nuclear Medicine	19/02/98	400	F

Write SQL commands for (a) to (f) and write output for (g).

(a) To show all information about the patients of Cardiology department.

(b) To list the names of female patients who are in Orthopedic department.

(c) To list names of all patients with their date of admission in ascending order.

(d) To display Patient's Name, Charges, Age for Male Patients only.

(e) To count the number of patients with Age > 30.

(f) To insert a new row in the 'Hospital' table with the following data :
11, "Mustafa", 37, "ENT", {25/02/98}, 250, "M"

(g) Give the output of the following SQL statements:

(i) Select COUNT (distinct Department) from Hospital;

(ii) Select MAX(Age) from Hospital where Sex = "M";

(iii) Select AVG(Charges) from Hospital where Sex = "F";

Select SUM(Charges) from Hospital where Dateofadm < {12/08/98};

SOLUTION:

(a) Select * from Hospital where Department = "Cardiology";

(b) Select Name from Hospital where Sex = "F" and Department = "Orthopedic";

(c) Select Name from Hospital order by Dateofadm;

(d) Select Name, Charges, Age from Hospital where Sex = "M";

(e) Select COUNT(*) from Hospital where Age > 30;

(f) Insert into Hospital Values (11, "Mustafa", 37, "ENT", {25/02/98}, 250, "M");

(g) **(i)** 6 **(ii)** 65
(iii) 400 **(iv)** 3850

3. Given the following tables for a database Library:

Table : Books

Book_Id	Book_Name	Author_Name	Publishers	Price	Type	Qty.
C0001	Fast Cook	Lata Kapoor	EPB	355	Cookery	5
F0001	The Tears	William Hopkine	First Publ.	650	Fiction	20
T0001	My First C++	Brian & Brooke	EPB	350	Text	10
T0002	C++ Brainworks	A.W. Rossaine	TDH	350	Text	15
F0002	Thunderbolts	Anna Roberts	First Publ.	750	Fiction	50

Book_Id	Quantity_Issued
T0001	4
C0001	5
F0001	2

Write SQL commands for (a) to (f) and write output for (g).

(a) To show Book name, Author name and Price of Books of First Publ. publishers.

(b) To list the names from books of Text type.

(c) To display the names and price from Books in ascending order of their price.

(d) To increase the price of all books of EPB Publishers by 50.

(e) To display the Book_Id, Book_Name and Quantity_Issued for all books which have been issued. (The query will require contents from both the tables.)

(f) To insert a new row in the table Issued having the following data : "F0003", 1

(g) Give the output of the following SQL statements:

(i) Select COUNT (*) from Books;

(ii) Select MAX(Price) from Books where Quantity >= 15;

(iii) Select Book_Name, Author_Name from Books where Publishers = "EPB";

(iv) Select COUNT(DISTINCT Publishers) from Books where Price >= 400;

SOLUTION:

(a) Select Book_Name, Author_Name, Price from Books where Publishers = "First Publ.";

(b) Select Book_Name from Books where Type = "Text";

(c) Select Book_Name, Price from Books order by Price;

(d) Update Books Set Price = Price + 50 where Publishers = "EPB";

(e) Select Books.Book_Id, Book_Name, Quantity_Issued from Books, Issued where Books.Book_Id = Issued.Book_Id

(f) Insert into Issued Values ("F0003", 1);

(g) **(i)** 5 **(ii)** 65

(iii) Fast Cook Lata Kapoor

My First C++ Brian & Brooke

(iv) 3850

SQL COMMANDS (Practice)

Consider the tables given below and answer the questions that follow:

Table: Employee

No	Name	Salary	Zone	Age	Grade	Dept
1	Mukul	30000	West	28	A	10
2	Kritika	35000	Centre	30	A	10
3	Naveen	32000	West	40		20
4	Uday	38000	North	38	C	30
5	Nupur	32000	East	26		20
6	Moksh	37000	South	28	B	10
7	Shelly	36000	North	26	A	30

Table: Department

Dept	DName	MinSal	MaxSal	HOD
10	Sales	25000	32000	1
20	Finance	30000	50000	5
30	Admin	25000	40000	7

Write SQL commands to:

Create Table / Insert Into

1. Create the above tables and insert tuples in them.

Simple Select

2. Display the details of all the employees.

3. Display the Salary, Zone, and Grade of all the employees.

4. Display the records of all the employees along with their annual salaries. The Salary column of the table contains monthly salaries of the employees.

5. Display the records of all the employees along with their annual salaries. The Salary column of the table contains monthly salaries of the employees. The new column should be given the name "Annual Salary".

Conditional Select using Where Clause

6. Display the details of all the employees who are below 30 years of age.

7. Display the names of all the employees working in North zone.

8. Display the salaries of all the employees of department 10.

Using NULL

9. Display the details of all the employees whose Grade is NULL.

10. Display the details of all the employees whose Grade is not NULL.

Using DISTINCT Clause

11. Display the names of various zones from the table Employee. A zone name should appear only once.

12. Display the various department numbers from the table Employee. A department number should be displayed only once.

Using Logical Operators (NOT, AND, OR)

13. Display the details of all the employees of department 10 who are above 30 years of age.

14. Display the details of all the employees who are getting a salary of more than 35000 in the department 30.

15. Display the names and salaries of all the employees who are working neither in West zone nor in Centre zone.

16. Display the names of all the employees who are working in department 20 or 30.

17. Display the details of all the employees whose salary is between 32000 and 38000.

18. Display the details of all the employees whose grade is between 'A' and 'C'

Using IN Operator

19. Display the names of all the employees who are working in department 20 or 30. (Using IN operator)

20. Display the names and salaries of all the employees who are working neither in West zone nor in Centre zone. (Using IN operator)

Using BETWEEN Operator

21. Display the details of all the employees whose salary is between 32000 and 38000. (Using BETWEEN operator)

22. Display the details of all the employees whose grade is between 'A' and 'C' (Using BETWEEN operator)

Using LIKE Operator

23. Display the name, salary, and age of all the employees whose names start with 'M'.

24. Display the name, salary, and age of all the employees whose names end with 'a'.

25. Display the name, salary, and age of all the employees whose names contain 'a'

26. Display the name, salary, and age of all the employees whose names do not contain 'a'

27. Display the details of all the employees whose names contain 'a' as the second character.

Using Aggregate functions

28. Display the sum and average of the salaries of all the employees.

29. Display the highest and the lowest salaries being paid in department 10.

30. Display the number of employees working in department 10.

Using ORDER BY clause

31. Display the details of all the employees in the ascending order of their salaries.

32. Display the details of all the employees in the descending order of their names.

33. Display the details of all the employees in the ascending order of their grades and within grades in the descending order of their salaries.

Using GROUP BY clause

34. Display the total number of employees in each department.

35. Display the highest salary, lowest salary, and average salary of each zone.

36. Display the average age of employees in each department only for those departments in which average age is more than 30.

Using UPDATE, DELETE, ALTER TABLE

37. Put the grade B for all those whose grade is NULL.

38. Increase the salary of all the employees above 30 years of age by 10%.

39. Delete the records of all the employees whose grade is C and salary is below 30000.

40. Delete the records of all the employees of department 10 who are above 40 years of age.

41. Add another column HireDate of type Date in the Employee table.

JOIN of two tables

42. Display the details of all the employees who work in Sales department.

43. Display the Name and Department Name of all the employees.

DROP TABLE

44. Drop the tables Employee and Department.

```
mysql> use class12;
  Database changed
```

Create Table / Insert Into

1. Create the above tables and insert tuples in them.

```
mysql> create table employee(no int(2),name varchar(30),salary int(8),zone
      varchar(10),age int(3), grade varchar(5),dept int(3));
Query OK, 0 rows affected (0.09 sec)

mysql>insert into employee values(1,'mukul',30000,'west',28,'A',10);
Query OK, 1 row affected (0.14 sec)
mysql>insert into employee values(2,'kritika',35000,'centre',30,'A',10);
Query OK, 1 row affected (0.12 sec)
mysql>insert into employee values(3,'naveen',32000,'west',40,NULL,20);
Query OK, 1 row affected (0.1 sec)
mysql>insert into employee values(4,'uday',38000,'north',38,'C',30);
Query OK, 1 row affected (0.18 sec)
mysql>insert into employee values(5,'nupur',32000,'east',26,NULL,20);
Query OK, 1 row affected (0.2 sec)
mysql>insert into employee values(6,'moksh',37000,'south',28,'B',10);
Query OK, 1 row affected (0.13 sec)
mysql>insert into employee values(7,'shelly',36000,'north',26,'A',30);
Query OK, 1 row affected (0.14 sec)

mysql>create table department(dept int(3), dname varchar(20), minsal int(8), maxsal
      int(8), HOD int(2));
```

```
Query OK, 0 row affected (0.1 sec)
mysql>insert into department values(10,'sales',25000,32000,1);
Query OK, 1 row affected (0.13 sec)
mysql>insert into department values(20,'finance',30000,50000,5);
Query OK, 1 row affected (0.14 sec)
mysql>insert into department values(30,'admin',25000,40000,7);
Query OK, 1 row affected (0.11 sec)


mysql> show tables;
+------------------+
| Tables_in_chiya  |
+------------------+
| department       |
| employee         |
+------------------+
2 rows in set (0.02 sec)
```

Simple Select

2. Display the details of all the employees.

```
mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  32000 | west   |   40 | NULL  |   20 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.06 sec)

mysql> select *from department;
+------+---------+--------+--------+------+
| dept | dname   | minsal | maxsal | HOD  |
+------+---------+--------+--------+------+
|   10 | sales   |  25000 |  32000 |    1 |
|   20 | finance |  30000 |  50000 |    5 |
|   30 | admin   |  25000 |  40000 |    7 |
+------+---------+--------+--------+------+
3 rows in set (0.02 sec)
```

3. Display the salary, zone, and grade of all the empolyees.

```
mysql> select salary,zone,grade from employee;
+--------+--------+-------+
| salary | zone   | grade |
+--------+--------+-------+
|  30000 | west   | A     |
|  35000 | centre | A     |
|  32000 | west   | NULL  |
|  38000 | north  | C     |
|  32000 | east   | NULL  |
|  37000 | south  | B     |
|  36000 | north  | A     |
+--------+--------+-------+
7 rows in set (0.00 sec)
```

4. Display the records of all the employees along with their annual salaries. The Salary column of the table contains monthly salaries of the employees.

```
mysql> select no, name, zone, age, grade, dept, salary*12 'annual salary' from
      employee;
+------+---------+--------+------+-------+------+---------------+
| no   | name    | zone   | age  | grade | dept | annual salary |
+------+---------+--------+------+-------+------+---------------+
|    1 | mukul   | west   |   28 | A     |   10 |        360000 |
|    2 | kritika | centre |   30 | A     |   10 |        420000 |
|    3 | naveen  | west   |   40 | NULL  |   20 |        384000 |
|    4 | uday    | north  |   38 | C     |   30 |        456000 |
|    5 | nupur   | east   |   26 | NULL  |   20 |        384000 |
|    6 | moksh   | south  |   28 | B     |   10 |        444000 |
|    7 | shelly  | north  |   26 | A     |   30 |        432000 |
+------+---------+--------+------+-------+------+---------------+
7 rows in set (0.00 sec)
```

Conditional Select using Where Clause

6. Display the details of all the employees who are below 30 years of age.

```
mysql> select * from employee where age<30;
+------+--------+--------+-------+------+-------+------+
| no   | name   | salary | zone  | age  | grade | dept |
+------+--------+--------+-------+------+-------+------+
|    1 | mukul  |  30000 | west  |   28 | A     |   10 |
|    5 | nupur  |  32000 | east  |   26 | NULL  |   20 |
|    6 | moksh  |  37000 | south |   28 | B     |   10 |
|    7 | shelly |  36000 | north |   26 | A     |   30 |
+------+--------+--------+-------+------+-------+------+
4 rows in set (0.02 sec)
```

7. Display the names of all the employees working in North zone.

```
mysql> select name from employee where zone='north';
+--------+
| name   |
+--------+
| uday   |
| shelly |
+--------+
2 rows in set (0.00 sec)
```

8. Display the salaries of all the employees of department 10.

```
mysql> select salary from employee where dept=10;
+--------+
| salary |
+--------+
|  30000 |
|  35000 |
|  37000 |
+--------+
3 rows in set (0.00 sec)
```

Using NULL

9. Display the details of all the employees whose Grade is NULL.

```
mysql> select * from employee where grade is null;
+------+--------+--------+------+------+-------+------+
| no   | name   | salary | zone | age  | grade | dept |
+------+--------+--------+------+------+-------+------+
|    3 | naveen |  32000 | west |   40 | NULL  |   20 |
|    5 | nupur  |  32000 | east |   26 | NULL  |   20 |
+------+--------+--------+------+------+-------+------+
2 rows in set (0.00 sec)
```

10. Display the details of all the employees whose Grade is not NULL.

```
mysql> select * from employee where grade is not null;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
5 rows in set (0.00 sec)
```

Using DISTINCT Clause

11. Display the names of various zones from the table Employee. A zone name should appear only once.

```
mysql> select distinct zone from employee;
+--------+
| zone   |
+--------+
| west   |
| centre |
| north  |
| east   |
| south  |
+--------+
5 rows in set (0.00 sec)
```

12. Display the various department numbers from the table Employee. A department number should be displayed only once.

```
mysql> select distinct dept from employee;
+------+
| dept |
+------+
|   10 |
|   20 |
|   30 |
+------+
3 rows in set (0.00 sec)
```

Using Logical Operators (NOT, AND, OR)

13. Display the details of all the employees of department 10 who are above 30 years of age.

```
mysql> select * from employee where dept=10 and age>=30;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
+------+---------+--------+--------+------+-------+------+
1 row in set (0.00 sec)
```

14. Display the details of all the employees who are getting a salary of more than 35000 in the department 30.

```
mysql> select * from employee where salary>35000 and dept=30;
+------+--------+--------+-------+------+-------+------+
| no   | name   | salary | zone  | age  | grade | dept |
+------+--------+--------+-------+------+-------+------+
|    4 | uday   |  38000 | north |   38 | C     |   30 |
|    7 | shelly |  36000 | north |   26 | A     |   30 |
+------+--------+--------+-------+------+-------+------+
2 rows in set (0.00 sec)
```

15. Display the names and salaries of all the employees who are working neither in West zone nor in Centre zone.

```
mysql> select name, salary from employee where zone!='west' and zone!='centre';
+--------+--------+
| name   | salary |
+--------+--------+
| uday   |  38000 |
| nupur  |  32000 |
| moksh  |  37000 |
| shelly |  36000 |
+--------+--------+
4 rows in set (0.00 sec)
```

16. Display the names of all the employees who are working in department 20 or 30.

```
mysql> select name from employee where dept=20 or dept=30;
+--------+
| name   |
+--------+
| naveen |
| uday   |
| nupur  |
| shelly |
+--------+
4 rows in set (0.00 sec)
```

17. Display the details of all the employees whose salary is between 32000 and 38000.

```
mysql> select * from employee where salary>32000 and salary<38000;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
3 rows in set (0.00 sec)
```

18. Display the details of all the employees whose grade is between 'A' and 'C'.

```
mysql> select * from employee where grade between 'A' and 'C';
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
5 rows in set (0.00 sec)
```

19. Display the names of all the employees who are working in department 20 or 30.

```
mysql> select name from employee
    -> where dept in(20,30);
```

```
+--------+
| name   |
+--------+
| naveen |
| uday   |
| nupur  |
| shelly |
+--------+
4 rows in set (0.00 sec)
```

20. Display the names and salaries of all the employees who are working neither in West zone nor in Centre zone.

```
mysql> select name,salary from employee
    -> where zone not in('centre','west');
+--------+--------+
| name   | salary |
+--------+--------+
| uday   |  38000 |
| nupur  |  32000 |
| moksh  |  37000 |
| shelly |  36000 |
+--------+--------+
4 rows in set (0.00 sec)
```

21. Display the details of all the employees whose salary is between 32000 and 38000

```
mysql> select * from employee where salary between 32000 and 38000;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  32000 | west   |   40 | NULL  |   20 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
6 rows in set (0.01 sec)
```

22. Display the details of all the employees whose grade is between ‘A’ and ’C’.

```
mysql> select * from employee where grade between 'A' and 'C';
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
5 rows in set (0.00 sec)
```

Using LIKE Operator

23. Display the name, salary, and age of all the employees whose names start with 'M'.

```
mysql> select name, salary, age from employee where name like 'm%';
+-------+--------+------+
| name  | salary | age  |
+-------+--------+------+
| mukul |  30000 |   28 |
| moksh |  37000 |   28 |
+-------+--------+------+
2 rows in set (0.00 sec)
```

24. Display the name, salary, and age of all the employees whose names end with 'a'.

```
mysql> select name, salary, age from employee where name like '%a';
+---------+--------+------+
| name    | salary | age  |
+---------+--------+------+
| kritika |  35000 |   30 |
+---------+--------+------+
1 row in set (0.00 sec)
```

25. Display the name, salary, and age of all the employees whose names contain 'a'.

```
mysql> select name, salary, age from employee where name like '%a%';
+---------+--------+------+
| name    | salary | age  |
+---------+--------+------+
| kritika |  35000 |   30 |
| naveen  |  32000 |   40 |
| uday    |  38000 |   38 |
+---------+--------+------+
3 rows in set (0.00 sec)
```

26. Display the name, salary, and age of all the employees whose names do not contain 'a'.

```
mysql> select name, salary, age from employee where name not like '%a%';
+--------+--------+------+
| name   | salary | age  |
+--------+--------+------+
| mukul  |  30000 |   28 |
| nupur  |  32000 |   26 |
| moksh  |  37000 |   28 |
| shelly |  36000 |   26 |
+--------+--------+------+
4 rows in set (0.00 sec)
```

27. Display the details of all the employees whose names contain 'a' as the second character.

```
mysql> select name, salary, age from employee where name like'_a%';
+--------+--------+------+
| name   | salary | age  |
+--------+--------+------+
| naveen |  32000 |   40 |
+--------+--------+------+
1 row in set (0.00 sec)
```

Using Aggregate functions

28. Display the sum and average of the salaries of all the employees.

```
mysql> select sum(salary) 'total salary', avg(salary) 'average salary' from
      employee;
+--------------+----------------+
| total salary | average salary |
+--------------+----------------+
|       240000 |     34285.7143 |
+--------------+----------------+
1 row in set (0.03 sec)
```

29. Display the highest and the lowest salaries being paid in department 10.

```
mysql> select min(salary), max(salary) from employee where dept=10;
+-------------+-------------+
| min(salary) | max(salary) |
+-------------+-------------+
|       30000 |       37000 |
+-------------+-------------+
1 row in set (0.02 sec)
```

Using ORDER BY clause

31. Display the details of all the employees in the ascending order of their salaries.

```
mysql> select * from employee order by salary;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    3 | naveen  |  32000 | west   |   40 | NULL  |   20 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

32. Display the details of all the employees in the descending order of their names.

```
mysql> select * from employee order by name desc;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    3 | naveen  |  32000 | west   |   40 | NULL  |   20 |
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.02 sec)
```

33. Display the details of all the employees in the ascending order of their grades and within grades in the descending order of their salaries.

```
mysql> select * from employee order by grade asc, salary desc;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    3 | naveen  |  32000 | west   |   40 | NULL  |   20 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

Using GROUP BY clause

34. Display the total number of employees in each department.

```
mysql> select dept ,count(*) from employee group by dept;
+------+----------+
| dept | count(*) |
+------+----------+
|   10 |        3 |
|   20 |        2 |
|   30 |        2 |
+------+----------+
3 rows in set (0.03 sec)
```

35. Display the highest salary, lowest salary, and average salary of each zone.

```
mysql> select min(salary)'minimum salary' , max(salary)'maximum salary' ,
      avg(salary)'average salary' from employee group by zone;
+----------------+----------------+----------------+
| minimum salary | maximum salary | average salary |
+----------------+----------------+----------------+
|    35000       |     35000      |   35000.0000   |
|    32000       |     32000      |   32000.0000   |
|    36000       |     38000      |   37000.0000   |
|    37000       |     37000      |   37000.0000   |
|    30000       |     32000      |   31000.0000   |
+----------------+----------------+----------------+
5 rows in set (0.02 sec)
```

36. Display the average age of employees in each department only for those departments in which average age is more than 30.

```
mysql> select dept,avg(age)'average age'from employee group by dept having
      avg(age)>30;
+------+-------------+
| dept | average age |
+------+-------------+
|   20 |     33.0000 |
|   30 |     32.0000 |
+------+-------------+
2 rows in set (0.00 sec)
```

Using UPDATE, DELETE, ALTER TABLE

37. Put the grade B for all those whose grade is NULL.

```
mysql> update employee set grade='B' where grade is null;
Query OK, 2 rows affected (0.06 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  32000 | west   |   40 | B     |   20 |
|    4 | uday    |  38000 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | B     |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

38. Increase the salary of all the employees above 30 years of age by 10%.

```
mysql> update employee set salary=(11/10*salary) where age>30;
Query OK, 2 rows affected (0.05 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  35200 | west   |   40 | NULL  |   20 |
|    4 | uday    |  41800 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

39. Delete the records of all the employees whose grade is C and salary is below 30000.

```
mysql> delete from employee where grade='C' and salary<30000;
Query OK, 0 rows affected (0.05 sec)
mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  35200 | west   |   40 | NULL  |   20 |
|    4 | uday    |  41800 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.02 sec)
```

40. Delete the records of all the employees of department 10 who are above 40 years of age.

```
mysql> delete from employee where dept=10 and age>=40;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+
| no   | name    | salary | zone   | age  | grade | dept |
+------+---------+--------+--------+------+-------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 |
|    3 | naveen  |  35200 | west   |   40 | NULL  |   20 |
|    4 | uday    |  41800 | north  |   38 | C     |   30 |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 |
+------+---------+--------+--------+------+-------+------+
7 rows in set (0.00 sec)
```

41. Add another column HireDate of type Date in the Employee table.

```
mysql> alter table employee add (Hiredate date);
Query OK, 7 rows affected (0.22 sec)
Records: 7  Duplicates: 0  Warnings: 0

mysql> select * from employee;
+------+---------+--------+--------+------+-------+------+----------+
| no   | name    | salary | zone   | age  | grade | dept | Hiredate |
+------+---------+--------+--------+------+-------+------+----------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 | NULL     |
|    2 | kritika |  35000 | centre |   30 | A     |   10 | NULL     |
|    3 | naveen  |  35200 | west   |   40 | NULL  |   20 | NULL     |
|    4 | uday    |  41800 | north  |   38 | C     |   30 | NULL     |
|    5 | nupur   |  32000 | east   |   26 | NULL  |   20 | NULL     |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 | NULL     |
|    7 | shelly  |  36000 | north  |   26 | A     |   30 | NULL     |
+------+---------+--------+--------+------+-------+------+----------+
7 rows in set (0.00 sec)
```

JOIN of two tables

42. Display the details of all the employees who work in Sales department.

```
mysql> select * from employee, department where employee.dept=10 and
employee.dept=department.dept;
+------+---------+--------+--------+------+-------+------+----------+------+-------
      +--------+--------+------+
| no   | name    | salary | zone   | age  | grade | dept | Hiredate | dept | dname |
      minsal | maxsal | HOD  |
+------+---------+--------+--------+------+-------+------+----------+------+-------
      +--------+--------+------+
|    1 | mukul   |  30000 | west   |   28 | A     |   10 | NULL     |   10 | sales |
      25000 |  32000 |    1 |
|    2 | kritika |  35000 | centre |   30 | A     |   10 | NULL     |   10 | sales |
      25000 |  32000 |    1 |
|    6 | moksh   |  37000 | south  |   28 | B     |   10 | NULL     |   10 | sales |
      25000 |  32000 |    1 |
+------+---------+--------+--------+------+-------+------+----------+------+-------
      +--------+--------+------+
3 rows in set (0.00 sec)
```

43. Display the Name and Department Name of all the employees.

```
mysql> select dname, name from department, employee where
      department.dept=employee.dept;
+---------+---------+
| dname   | name    |
+---------+---------+
| sales   | mukul   |
| sales   | kritika |
| finance | naveen  |
| admin   | uday    |
| finance | nupur   |
| sales   | moksh   |
| admin   | shelly  |
+---------+---------+
7 rows in set (0.00 sec)
```

44. Drop tables

```
mysql> drop table department;
Query OK, 0 rows affected (0.12 sec)

mysql> drop table employee;
Query OK, 0 rows affected (0.09 sec)

mysql> exit;
```

Chapter - 8	Boolean Algebra

1. Problems based on Truth Table:

(a) Verify using truth table:

e.g. Verify X + Y.Z = (X + Y).(X + Z) using truth table.

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

$\therefore$ On comparing columns X + Y.Z and (X + Y).(X + Z) we obtain both these columns are equal.

Hence proved.

(b). Writing POS or SOP form for a given truth table.

e.g. Write POS and SOP form of:-

	U	V	W	F
(i).	0	0	0	1
(ii).	0	0	1	0
(iii).	0	1	0	1
(iv).	0	1	1	0
(v)	1	0	0	1
(vi)	1	0	1	0
(vii)	1	1	0	1
(viii)	1	1	1	1

POS ÷ (Write terms involving 0 as the output.)

$F = \left(U + V + \overline{W}\right).\left(U + \overline{V} + \overline{W}\right).\left(\overline{U} + V + \overline{W}\right)$

SOP ÷ (Write terms involving 1 as the output.)

$F = \left(\overline{U}.\overline{V}.\overline{W}\right) + \overline{U}.V.\overline{W} + U.\overline{V}.\overline{W} + U.V.\overline{W} + U.V.W$

2. Problems based on duals and complements.

(a). Finding duals:-

(working rule)

$\rightarrow$ Replace (+) by (.) ; (.) by (+) ; (0) by (1); (1) by (0).

e.g. (i) $X + \overline{X}.Y$

$\rightarrow$ $X.\left(\overline{X} + Y\right)$

(ii) $(A + 0).(A.1.\bar{A})$

$\rightarrow$ $(A.1)+(A+0+\bar{A})$

(b) Finding complementary :
e.g. $A.B' + C'.D'$

$\rightarrow$ $(A.B' + C'.D')' = (A.B')'.(C'.D')'$ (Demorgan's first theorem $\overline{X+Y} = \bar{X}.\bar{Y}$)

$= (A'+ B").(C"+ D")$ (demorgans second theorem $\overline{X.Y}=\bar{X}+\bar{Y}$)

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

3. Problems based on POS and SOP form.

(a) Find minterm designation of $\bar{X}.Y.\bar{Z}$

$\rightarrow$ Binary equivalent = 0 1 0

Decimal equivalent = $0\times2^2 + 1\times2^1 + 0\times2^0 = 2$

$\therefore \bar{X}.Y.\bar{Z} = m_2$

(b) Find minterms of A.B + C

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

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

$A.B.C + A.B.\bar{C} + A.B.C + A.\bar{B}.C + \bar{A}.B.C + \bar{A}.\bar{B}.C$

$A.B.C + A.B.\bar{C} + A.\bar{B}.C + \bar{A}.B.C + \bar{A}.\bar{B}.C$ (Removing duplicate terms), this is the desired SOP form.

❖ If expressing is in complex form first simply it to the maximum and then proceed the same.

e.g. $\overline{(\bar{X}.Y)}\overline{(\bar{X}.\bar{Z})} = (\bar{\bar{X}}+\bar{Y})(\bar{\bar{X}}+\bar{\bar{Z}})$ $(\because \overline{A.B} = \bar{A}+\bar{B})$

$= (X+\bar{Y})(X+Z)$ $(\because \bar{\bar{A}} = A)$

$= X + \bar{Y}.Z$ $(\ X + Y.Z = (X+Y)(X+Z))$

(c). Find maxterms of $\bar{X}.Y + Y.\bar{Z}$

$\rightarrow$ $\bar{X}.Y + Y.\bar{Z} = (\bar{X}.Y+Y)(\bar{X}.Y+\bar{Z})$ $(\ A+BC = (A+B).(A+C))$

$= (\bar{X}+Y).(Y+Y).(\bar{Z}+Y).(\bar{Z}+\bar{X}) = (\bar{X}+Y).(Y).(\bar{Z}+Y).(\bar{Z}+\bar{X})$

$= (\bar{X}+Y+Z.\bar{Z}).(Y+X.\bar{X}+Z.\bar{Z}).(X.\bar{X}+\bar{Z}+Y).(Y.\bar{Y}+\bar{X}+\bar{Z})$

$= (\bar{X}+Y+Z).(\bar{X}+Y+\bar{Z}).(X.\bar{X}+Y+Z).(X.\bar{X}+Y+\bar{Z}).(Y+\bar{Z}+X)$

$(Y+\bar{Z}+\bar{X}).(\bar{Z}+\bar{X}+Y).(\bar{Z}+\bar{X}+\bar{Y})$

$(\because A+BC = (A+B).(A+C))$

$= (\bar{X}+Y+Z).(\bar{X}+Y+\bar{Z}).(Z+Y+X).(Z+Y+\bar{X}).(\bar{Z}+Y+X).(\bar{Z}+\bar{X}+\bar{X})$

$(X+\bar{X}+Y)(\bar{X}+\bar{X}+Y)(\bar{X}+\bar{X}+Z)(\bar{X}+\bar{Y}+\bar{Z})$

Desired POS form (after removing duplicate terms)

(d) Convert this POS to SOP form.

$(\bar{X}+Y+\bar{Z}).(\bar{X}+Y+Z).(\bar{X}+\bar{Y}+Z).(\bar{X}+\bar{Y}+\bar{Z})$

$\rightarrow$ $F = (\bar{X}+Y+\bar{Z}).(\bar{X}+Y+Z).(\bar{X}+\bar{Y}+Z).(\bar{X}+\bar{Y}+\bar{Z})$

$= (1\,0\,1)$ $(1\,0\,0)$ $(1\,1\,0)$ $(1\,1\,1)$

$= M_5 \quad M_4 \quad M_6 \quad M_7$

$= \pi(4, 5, 6, 7)$

Now, equivalent SOP expression will be;

$\sum(0,1,2,3)$

$= m_0 + m_1 + m_2 + m_3$

$(000)\ (001)\ (010)\ (011)$

$= \bar{X}.\bar{Y}.\bar{Z} + \bar{X}.\bar{Y}.Z + \bar{X}.Y.\bar{Z} + \bar{X}.Y.Z$

Define (i) minterms (ii) maxterms (iii) canonical form.

(i) A minterms is a product of all the literals (with or without bar) within the logic system e.g. $X.Y.Z + X.\bar{Y}.\bar{Z}$.....

(ii). A maxterm is a sum of all literals (with or without bar) within the logic system. e.g. $(X+Y+Z).(\bar{X}+\bar{Y}+\bar{Z})$....

(iii). A boolean expression composed entirely either of minterms or maxterms is referred as conical expression.

4. Laws :-

a). Associative law

(i) $X+(Y+Z)=(X+Y)+Z$ (ii) $X.(Y.Z)=(X.Y).Z$

b). Distributive law

(i) $X.(Y+Z)=X.Y+X.Z$ (ii) $X+Y.Z=(X+Y).(X+Z)$

Algebric verification:-

RHS $= X.X + X.Z + Y.X + Y.Z$

$= X + X.Z + X.Y + Y.Z \quad (A.A = A)$

$= X.1 + X.Y + Y.Z \quad (1 + A = 1)$

$= X.(1+Y) + Y.Z$

$= X.1 + Y.Z \quad (1 + A = 1)$

LHS $= X + Y.Z$

Hence verified.

c). Commutative law

(i) $X+Y=Y+X$ (ii) $X.Y=Y.X$

d). Involution law

$\bar{\bar{X}} = X$

e). Idempotent law

(i) $X+X=X$ (ii) $X.X=X$

f). Absortion law

(i) $X+X.Y=X$

Algebric verification:

LHS $= X + X.Y$

$= X.(1+Y)$

$= X.1 \quad (1 + A = 1)$

$= X \quad (1.A = A)$

RHS $= X$

Hence verified

(ii) $X + \overline{X}Y = X + Y$

Algebric verification:

LHS $= X + \overline{X}Y$

$= (X + \overline{X}).(X + Y)$ ($A + B.C = (A + B).(A + C)$)

$= 1.(X + Y)$

$= X + Y$ ($1.A = A$)

RHS $= X + Y$

Hence verified

g). Demoryan's theorem

(i) $\overline{X + Y} = \overline{X} \cdot \overline{Y}$ (ii) $\overline{X.Y} = \overline{X} + \overline{Y}$

Proof for Demorgan's Theorem : -

(i) $\overline{X + Y} = \overline{X}.\overline{Y}$

Let P = X + Y

Now, if $\overline{P} = \overline{X + Y} = \overline{X}.\overline{Y}$

Then $P + \overline{P} = 1 and\ P.\overline{P} = 0$

$P + \overline{P} = X + Y + \overline{X + Y}$

$= X + Y + \overline{X}.\overline{Y}$

$= (X + \overline{X}).(X + \overline{Y}) + Y$ (A + B.C = (A + B).(B + C))

$= 1.\left(X + \overline{Y}\right) + Y$ $\left(\because A + \overline{A} = 1\right)$

$= X + \overline{Y} + Y$ $\left(\because 1.A = A \& A + \overline{A} = 1\right)$

= X + 1 = 1 $\left(1 + A = 1\right)$

$P.\overline{P} = (X + Y).(\overline{X + Y})$

$= (X + Y).(\overline{X}.\overline{Y})$

$= X.\overline{X}.\overline{Y} + Y.\overline{Y}.\overline{X}$ ($A.(B + C) = A.B + A.C$)

$= 0.\overline{Y} + 0.\overline{X}$ ($A.\overline{A} = 0$)

$= 0 + 0$ ($0.A = 0$)

(ii) $\overline{X.Y} = \overline{X} + \overline{Y}$

Let $P = X.Y$

Now, if $\overline{P} = \overline{X.Y} = \overline{X} + \overline{Y}$

Them $P + \overline{P} = 1 \quad P.\overline{P} = 0$

$P + \overline{P}$ $= X.Y + \overline{X.Y}$

$= X.Y + \overline{X} + \overline{Y}$

$= (\overline{X} + X).(\overline{X} + Y) + \overline{Y}$ ($A + B.C = (A + B).(A + C)$)

$= 1.(\overline{X} + Y) + \overline{Y}$($A + \overline{A} = 1$)

$= \overline{X} + Y + \overline{Y}$($1.A = A$) $= \overline{X} + 1 = 1$ ($1 + A = 1$)

$P.\overline{P} = (X.Y).(\overline{X.Y})$

$= (X.Y).(\overline{X} + \overline{Y})$

$= X.Y.\overline{X} + X.Y.\overline{Y}$ ($A(B+C) = AB + AC$)

$= 0.Y + 0.X$ ($A.\overline{A} = 0$)

$= 0 + 0$ ($0.A = 0$)

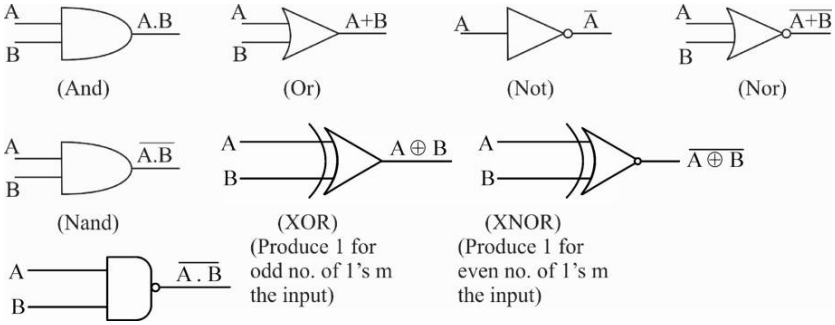
$= 0$

Q. How do distributive law here diffuse from that of ordinary algebra?

A $X.(Y+Z) = X.Y + X.Z$ holds good for all values of X,Y,Z in ordinary algebra.

But; $X + Y.Z = (X+Y).(X+Z)$ holds good only for two values $(0,1)$ of (X,Y,Z).

5. Problems based on logic gates.



(a) Draw circuit of given Boolean expression.

(i) Using basic logic gates (NOT, AND, OR)

(ii) Using NAND or NOR gate – (Draw using basic logic gates and replace each gate by NAND or NOR gate.)

(b) Write Boolean expression for given circuit.

Q Which gates are universal and why?

A NAND and NOR gates are universal as they are less expensive and easier to design. Also, other switching function (AND, OR) can easily be implemented using NAND/NOR gates)

6. Problems based an K-map:

1. For SOP expression.

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

* Use 1 to represent the desired boxed.

2. For POS expression.

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

* Use 0 to represent the desired boxed.

*Select only in multiple of 2, i.e. 2, 4, 8, 16…… in decreasing preference order.

Chapter - 9	Networking

- **Network**: Two or more computers are said to be in network if they can exchange data through some medium (wires, wireless etc.)
- Need for Networking :
  - **(i)** Resource Sharing
  - **(ii)** Reliability (if file on one system crashes, it could be retrieved from another PC)
  - **(iii)** Cost factor
  - **(iv)** Communication Medium
  - Application of Networking
  - **(i)** Sharing :
    - **(a)** Peripherals (Printers etc.)
    - **(b)** E-mails etc.
    - **(c)** Centrally controlled Network
  - **(ii)** Access to Remote Database
  - **(iii)** Communication Facilities
- **ARPANET** : Advanced Research Project Agency Network
- **NSFnet** : National Science Foundation network (more capable then ARPANET)
- **Internet** : World wide network of Computer Network.

❑ **ARPANET :**

- ARPANET (Advanced Research Projects agency Network) is first Network
- Developed in 1969 by U.S. Department Of Defense.
- Goal was to connect Computers at different Universities and U.S. Defense.
- Used by scientists, engineers, researchers and students for exchanging messages, data, share interests.

❑ **NSFnet :**

- High Capacity Network NSFnet developed in mid 80's.
- NSFnet developed by National Science Foundation.
- Used For Academic Research

❑ **Internet (World Wide Network Of Computers) :**

- Private Companies built their own Networks
- And this was termed as Internet
- Later connected with ARPANET and NSFnet
- ARPANET and NSFnet discontinued in 90's

Made up of many networks each run by a different company and interconnected

❑ **Interspace :**

- It is Client/Server Software
- Allows multiple users to communicate audio, video, text in dynamic 3D environments
- **Gateway** : Device that connects dissimilar networks.
- **Backbone** : Central interconnecting structure that connects one or more networks just like the trunk of tree or spine of a human being.
- **TCP** : Transmission Control Protocol
  Responsible for dividing the file/message into small packets and reassembling them.
- **IP** : Internet Protocol
  Responsible for handling the address of destination computer so that each packet is routed to its proper destination.

- **Inter Space** : Client/Server software program that allows multiple users to communicate online with real-time audio, video and text chat in dynamic 3D environments.
- **Server** : Computer that facilitates the sharing of data, software, and hardware resources on the network.
  (i) Dedicated Servers **(ii)** Non-Dedicated Servers

- **NIU** : Network Interface unit
  Interpreter that helps establish comm. between the server and workstations.
- **MAC Address** : Physical address assigned by NIC manufacturer.

Switching Techniques

Circuit Switching	Message Switching	Packet Switching
Establishes Physical Connection between processes The data is sent and received Reserves bandwidth in advance E.g. Telephone	Follows store and forward procedure Process continues until the data reaches its destination Sends the data to switching office, stored on the disk, which then sends it to next office No permanent physical connection establishes	Same as message switching with the only difference that there is upper limit on the data that can be sent Same as message switching with the only difference that there is upper limit on the data that can be sent Improved performance as data packets are stored on main memory

- **Twisted pair Cable :**
- It has two insulated copper wires, about 1mm in diameter.
- The wires are twisted together to reduce electrical interference called crosstalk
- The twisted pair cable is the oldest but still used media.
- The twisted pair cable is generally used for internal telephone wiring.
- Twisted pair cables can run longer distances without amplification

Advantages :
- Easy to install and maintain
- Low weight
- Less expensive

Disadvantages :
- High Attenuation and Noise
- Low bandwidth, so unsuitable for broadband applications

- **Co-Axial Cable :**
- Co-axial cable has a copper wire surround by insulting material, which is encased by conducting braided mesh.
- Conducting mesh is enclosed in protective plastic sheet.
- The co-axial cables are used by cable television networks

Advantages :
- Higher Bandwidth then Twisted pair cable.
- Can be used for broadband applications
- Data transmission characteristics better than twisted pair cable

Disadvantages :

- Expensive than twisted pair cable.

- **Optical Fibers :**
  - Glass fibers are used as media to transmit the data in the form of light waves.
  - The glass fibers are covered in plastic jackets.
  - At the transmitting end of optical fiber, the laser or light emitting diodes are used and the receiving end has detectors to detect the signal.

Advantages :

- Immune to electrical and magnetic interference
- High Transmission Capacity
- Used for broadband applications

Disadvantages :

- Expensive
- Connecting two fibers is difficult
- Installation is difficult

- **Micro Waves**
  - Wireless media, which uses the microwaves to transfer the data.
  - Microwaves travel in the straight line and cannot penetrate the metal structure.
  - Microwave transmission requires the sender and receiver top be within line of sight.
  - The Microwave transmission is Terrestrial and Satellite.

- **Terrestrial Microwave :**
  High towers are installed with transmitter which is accurately aligned to the direction of receiver on the other tower. Microwave media is much cheaper than the optical fiber.

- **Satellite Microwave :**
  It is used for long distance transmission. The earth station sends the signals to the satellite, which transmits it back to another earth station.

Advantages :

- Cheaper
- Ease of communication over oceans and difficult terrains

Disadvantages :

- Insecure communication
- Bandwidth allocation is limited
- Propagation is affected by weather like rain, thunderstorm.

- **Radio Wave :**
  - Radio frequencies are available to business, private citizens for carrying voice signals over 10 miles of distance.
  - It has a transmitter and receiver both having antennas.

Advantages :

- Cheaper
- Ease of communication over difficult terrains

Disadvantages :

- Insecure communication
- Propagation is prone to weather effects

- **Satellite :**
- Special case of microwave
- It is used for long distance transmission
- The earth station sends the signals to the satellite, which retransmits it back to another earth station after amplifying it.

Advantages :

- Large area coverage
- Can be used for intercontinental communication

Disadvantages :

- High investment cost
- Overcrowding of bandwidths

Infrared : Secure Transmission.

Laser : Point-to-point transmission, adversely affected by weather.

- **Data Channel :** Medium used to carry info/data from one point to another.
- **Baud :** Unit of measurement for the info carrying capacity of Communication Channel.
- **Data transfer rates :**

bps	⇒	Bits per second
Bps	⇒	Bytes per second
Kbps	⇒	Kilo bits per second
KBps	⇒	Kilo bytes per second

- **Bandwidth :** Width of allocated band of frequencies to a channel
- **LAN :** Local Area Network
  MAN : Metropolitan Area Network
  WAN : Wide Area Network

- **LAN (Local Area Network)**
  - Connects computer with in a building or an organization.
  - Speed from 0.2 to 100 Mb/sec
  - Owned by single organization**.**
  - e.g Network in an office
- **MAN (Metropolitan Area Network)**
  - Connects the computer with in a town or city.
  - Covers area up to 50 km.
  - Not generally owned by single organization.
  - e.g. Cable TV Network
- **WAN (Wide Area Network)**
  - Connects the computers at larger distances
  - Like states, countries and continents.
  - Not owned by single organization.
  - e.g. Internet

➢ **Topologies :**

♦ Pattern of connecting the computers.

♦ Factors on which topology chosen depends are:

➢ Cost minimize installation cost

➢ Flexibility: Easy to extend

➢ Reliability: Easy fault detection

Different Topologies are:

♦ Bus

♦ Star

♦ Ring

♦ Tree

❑ **STAR TOPOLOGY :** Consists of central node to which all other nodes are connected

Advantages :

- ➢ Ease of Service
- ➢ Centralized control/Problem diagnosis
- ➢ One device per connection
- ➢ Simple access protocols.

Disadvantages :

- ➢ Long cable length
- ➢ Central Node Dependency
- ➢ Difficult to expand

❑ **BUS TOPOLOGY :** Consists of single cable on which all the nodes are connected

Advantage :

- ➢ Short cable length
- ➢ Resilient Architecture
- ➢ Simple wiring layout
- ➢ Easy to extend

Disadvantage :

- ➢ Fault Diagnosis Difficult
- ➢ Nodes must be intelligent
- ➢ Fault Isolation Difficult
- ➢ Repeater Configuration

❑ **RING/CIRCULAR TOPOLOGY :** Nodes are connected in the form of ring. Each node is connected to two neighboring nodes.

Advantage :

- ➢ Short Cable Length
- ➢ No wiring closet space req.
- ➢ Suitable for Optics Fibres

Disadvantage :

- ➢ Node failure causes network failure
- ➢ Difficult to Diagnose faults
- ➢ Network re-config. is difficult

❑ **TREE TOPOLOGY** : Modified form of bus topology. Forms inverted tree like structure.

Advantage :

- ➢ Easy to extend i.e. new nodes can be added easily.
- ➢ Fault isolation is easy.

Disadvantage :

- ➢ If the root node fails, whole network is down.

- **Modem** : Modulator Demodulator
  Computer peripheral that allows to connect one Computer with other via telephone lines.
- Telephone lines carries the analog signals. Modem is a device that can convert the digital signals to analog signals.**(i.e. modulation) and** convert that signals back to digital **(i.e. demodulation)**
- Modem is the device that converts digital signals to analog and vice-versa.
- **RJ-45** : Registered Jack-45, it is 8 wired connector for connecting in Ethernet LAN
- Computers that are part of Ethernet, have to install a special card called **Ethernet Card**. It has connections for Co-axial (BNC), twisted (RJ 45) or optical Fiber (AUI)
- **Hub** : Hardware device used to connect several computers together. Hub ranges in size from four to several hundred ports

Three types of Hub :

- **Active Hub:** This type of Hub regenerates the signal before forwarding them.
- **Passive Hub:** This type of Hub combines the signals of different network segments and forwards without regenerating.
- **Intelligent Hub:** This type of Hub regenerates the signal as well as chooses the path to send the signals.
- **Switch** : Device used to segment networks into different sub-networks called LAN segments. Filters the data i.e. transforms the data for forwarding packets
- **Repeater :** Device that amplifies & restores signals for long distance transmission.
- **Bridge :** Network device that establishes an intelligent connection between two local networks with the same standard but diff. type of cables.
- **Router :** Network device to separate diff. segments is a network to improve performance and reliability.
  A Router works like a bridge but can handle different protocols.
- **NIC :** Network Interface Card.
- **Ethernet :** LAN Architecture by Xerox corp. in association with DEC & Intel.
- **Protocol :** Formal description of message formats & rules that 2 or more machines must follow to exchange those messages.
- **HTTP :** Hypertext Transfer Protocol
  Set of rules for transferring hypertext on www
- **FTP :** File Transfer Protocol
  Standard for exchange of files across Internet.
- **Datagram :** Collection of the data that is sent as a single message.
- **SLIP :** Serial Line Internet Protocol
  For delivering IP packets over dialup lines.
- **PPP :** Point to Point Protocol
  For transmitting IP packets over several lines.
- **GSM :** Global System for Mobile Communication.
- **SIM :** Subscriber Identification Module
- **TDMA :** Time Division Multiple Access
- **CDMA :** Code Division Multiple Access
- **WLL :** Wireless in Local Loop.
- **3G :** Third Generation of mobile technology.
- **UMTS :** Universal Mobile Telecommunications System
- **EDGE :** Enhanced Data rates for Global Evolution
- **Telnet :** It is an Internet utility that lets you log onto remote computer system.
- **URL :** Uniform Resource Locator
  Unique address of a web site.

- **Web Browser** is a www client that navigates through the World Wide Web and displays web pages.
- **Web Server** is a www server that responds to the requests made by web browsers
- An Internet address which is character based is called a **Domain Name**.
- **POP3 :** Post Office Protocol version 3
- **SMTP :** Simple Mail Transfer Protocol
- **NNTP :** Network News Transfer Protocol
- A location on a net server is called **Web Site**
- A document that uses HTTP is called a **Web Page**
- **Web Hosting :** Means of hosting web-server application on a computer system through which electronic content on the Internet is readily available to any web browser client.
- **Web 2.0** refers to added features and applications that make the web more interactive, support easy online info exchange and interoperability.
- **HTML :** Hypertext Markup Language
- **XML :** eXtensible Markup Language
- **DHTML :** Dynamic HTML
- **Script :** List of commands embedded in a web-page scripts are interpreted and executed by a certain program or scripting engine.

  Client Side Scripts : Downloaded at the client end
  Eg. : VB Script, Java Script, Hypertext Preprocessor (PHP)
  Server Side Scripts : Enables completion/carrying out a task at server end and sending the results to client-end.
  Eg. : JSP (Java Server Pages), ASP (Active Server Pages)
  Perl, PHP (same as the of Client side)
- **OSS :** Open Source Software
- **FOSS :** Free and Open Source Software
- **FLOSS :** Free Livre and open source software
  Both free and open source software
- **GNU :** GNU's Not Unix
  GNU Project emphasizes on Freedom.
- **FSF :** Free software foundation
- **OSI :** Open source Initiative
- **W3C :** World wide web consortium
  Responsible for producing the software standards of www
- **Proprietory Software :** Neither open nor freely available software.
- **Freeware :** Free of Cost, Allows copying and further distribution BUT not modification
  Eg.: Microsoft Internet Explorer.
- **Shareware** :
  (i) Source code is not available
  (ii) Modifications to the software not allowed
- **Cookies :** 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 web site.
- **Crackers :** Malicious programmers who break into secure systems.
  Hackers : Interested in gaining knowledge about computer system and possibly using this knowledge for playful pranks.
- **Computer Virus :** Malicious program that requires a host and is designed to make a system sick, just like real virus.
- **Trojan Horse :** Code hidden in a program such as a game or spreadsheet that looks safe to run but has hidden side effects.

- **Worm** is a program designed to replicate.
- **Spam :** Electronic Junk Mail/Newsgroup postings.
- **ICMP :** Internet control message protocol.
- **IMAP :** Internet mail access protocol.
- **VoIP :** Voice over IP refers to a way to carry telephone calls over an IP data network. It offers a set of facilities to manage the delivery of voice info over Internet in digital form.
- **NFS :** Network File System.

❑ **TCP/IP**

- It is set of protocols that govern how data should flow across the network.
- Internet is based on TCP/IP
- It is the standard for the majority of networks

It has only four layers:

- Application layer
- Transport layer
- Internet layer
- Network Access layer
- TCP: Breaks the data into packets, verifies it and reassembles.
- IP: Envelopes the addresses, forwards to data to destination.

❑ **FTP(File Transport Protocol)**

- This protocol which enables files to be transferred between computers.
- Files of any type can be transferred.
- Works on client/server process.
- Files are transferred in compressed mode.

❑ **PPP (Point to Point Protocol)**

- Transmits IP packets over serial lines e.g. telephones
- Enables home users to avail internet access for their own PC.
- It handles error detection, supports multiple protocols, permits authentication.
- Enables to run GUI based browser, ftp client for ones PC.

❑ **Wireless/Mobile Computing**

- Wireless is transferring the information between a computing device, without the use of landlines. It involves cellular phones, laser or satellite communication.
- Mobile computing means computing device which is not always connected to central network like PDA, cell phones , laptop.

❑ **GSM**

- GSM Stands for Global System for Mobile communications
- GSM is fully digital system.
- This technique uses narrowband TDMA, which allows eight simultaneous calls on the same radio frequency.

❑ **SIM (Subscribers identity module)**

- It is a chip that gives cellular device its unique phone no.
- It has memory for storing data and applications, processor, and ability to interact with the user.
- Used in GSM mobile phones

- **CDMA (Code division Multiple Access)**
  - Every channel uses full available spectrum.
  - The conversations are encoded with a pseudo random digital sequence.
  - Each users signal is spread over entire bandwidth.
  - The signals are spread over the spectrum with Unique spreading code, which is also used at the receivers end for recovering the data.
- **WLL (Wireless in Local Loop)**
  - It connects subscribers to the Public switched telephone network using radio signals as a substitute for other connecting media.
  - WLL uses advanced transmission techniques that permits support a large subscriber bases.
- **3G and EDGE**
  - Analog cellular were first generation mobile phones, digital were the second generation.
  - 3G is third generation mobile phones, that work over wireless air interfaces such as GSM, CDMA.
  - *Enhanced Data Rates For Global Evolution i.e EDGE* is radio based high speed mobile data standard to meet the requirements of 3G.
- **EDGE**
  - *Enhanced Data Rates For Global Evolution i.e EDGE* is radio based high speed mobile data standard to meet the requirements of 3G.
  - A broadband packet based transmission of text, digitized voice, video, and multimedia
  - Data rates up to and possibly higher than 2Mbps
  - Offers set of services to mobile computers and phones anywhere in the world.
- **Web Hosting :** Means hosting the web server application on a computer system through which electronic content on the internet is available to web browser client.

Various Web hosting services are:

- Free Hosting
- Virtual or Shared Hosting
- \➢ Dedicated Hosting
- Collocation Hosting

Dedicated Hosting

- Company wishing to go online, rents an entire web server from hosting company.
- It is for large, high traffic sites with special needs as ecommerce.

Co- Location

- Company owns the server on which the site is hosted.
- Company owning the site is responsible for all server administration.

Free Hosting

- Web Pages are hosted for no cost.
- E.g. geocities, tripod, homestead etc.

Virtual Hosting

- This hosting is provided under one's own domain e.g http://www.yourname.com/www.yourname.com
- Access and update to the site and its file are secured.

HTML (Hyper Text Markup Language)

- Used for writing web pages
- This language tells the web browsers how to display the text , pictures, and links on the screen.
- It provides various tags for alignment, headings , lines, hyper linking etc.

XML (eXtensible Markup Language)

- It is a language for creating documents containing structured information.
- XML specification defines a standard way to add markup to documents.

Difference in HTML and XML

- In HTML, both tag and semantics are fixed but not in XML.
- XML is meta language, it provides facility to tags and structural relationship between them.

DHTML (Dynamic HTML)

- Refers to new tags that will enable a web page to react to users input without sending requests to the web server.
- It allows the web page to change after it is loaded into the web browser.
- It is animated HTML.

Network Security Concepts

- To make sure that only legal or authorized users can gain access to information.
- Problems encountered under network security are:
- Physical security holes
- Software security Holes
- Inconsistent Usage Holes
- Physical security holes:
- Individual gains unauthorized access to computer.
- Software security Holes:
- Badly written programs or privileged s/w are compromised into doing things that it should not do.
- Inconsistent Usage Holes:
- System is seriously flawed from security point of view while combination of H/w and S/w.

Protection Methods:

- Authorization: It confirms that the service requester is entitled to perform the operation or not. Login id is provided for this
- Authentication: ensures each entity in web service is what it actually claims to be. It is Password protection.

Encrypted Smart Cards

- It is hand held smart card that can generate a token for a computer to understand. Every time new token is generated.

Biometric Systems :

Some unique aspect of human body such as finger prints, retinal patterns etc. is used to establish identity.

Firewall :

- It is a system designed to prevent unauthorized access to or from a private network.
- Can be implemented on both Hardware and Software.
- Used to prevent Internet users to access Internet.
- It also protects external systems against attacks originating from your network.

Cookies

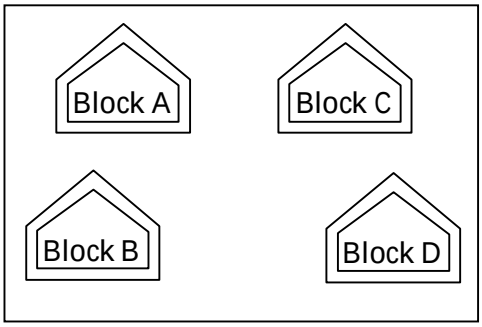
- Cookies are messaging that a web server transmits to a web server so that the web server can keep track of users activity on a specific web site.
- Its main purpose is to identify the user and prepare customized web pages.
- Cookies cannot read hard drive, cannot be used to prevent viruses.

The personal info. like credit card nos. etc. will be stored on the cookie unless you turn cookie off.

Client Side Scripting	Server Side Scripting
Script Code is downloaded and executed at client end.	The script is executed at the server-end and the result is sent to the client end.
Response to interaction is more immediate once the program code has been downloaded.	Complex Processes are more efficient as the program and associated resources are not downloaded to the browser.
Services are secure as they do not have access to files and databases.	Have access to files and databases but have security considerations when sending sensitive information.
Browser dependent.	Does not depend on browsers.
Affected by the processing speed of user's computer.	Affected by the processing speed of host server.

Long Answer Questions:

1. 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:



Center to centre distances between varius block

Block 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

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

(i) Suggest a cable layout conections between the blocks.

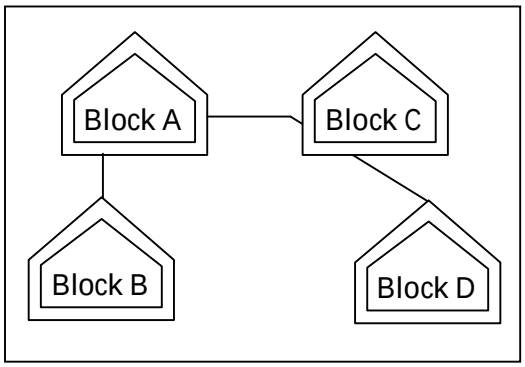
(ii) Suggest the most suitable place (i. e., block) to house the server of this organisation with a suitable reason.

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

(a) Repeater (b) Hub/Switch

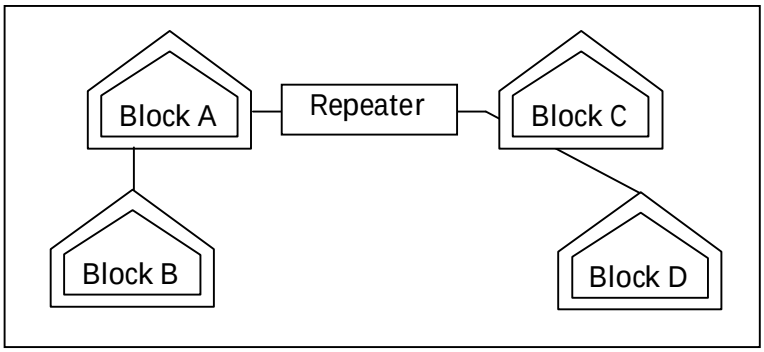
(iv) The organization is planning to link its front office situated in the city in a hilly region where cable connection is not feasible, suggest, suggest an economic way to connect it with reasonably high speed.

Sol. (i)

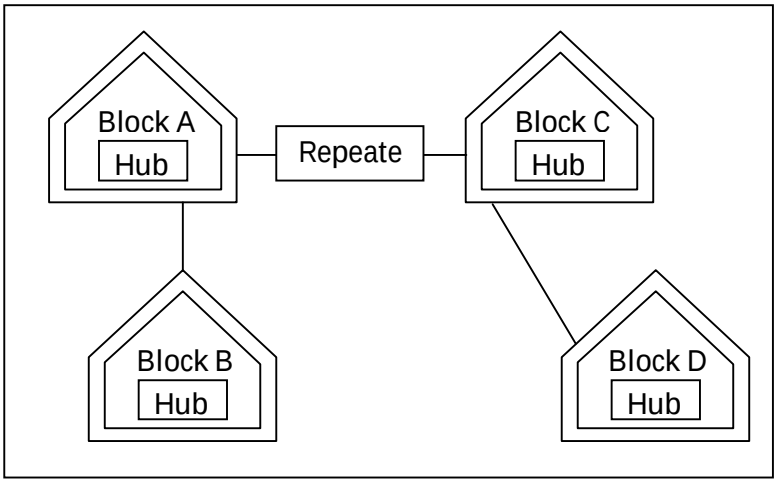


(ii) Block C. The most suitable plane/block to house the server of this organisation would be Block C, as this block contains the maximum number of computers, thus decreasing the cabling cost for most of the computers as well as increasing the efficiency of the maximum computers in the network.

(iii) (a)

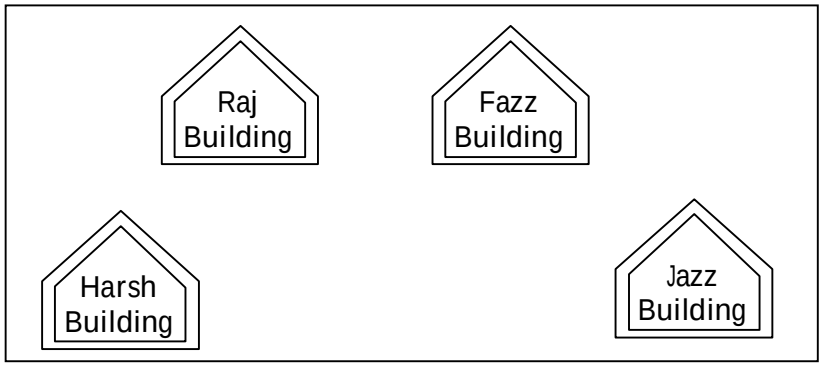


(b) In both the layouts, a hub/switch each would be needed in all the blocks, to interconnect the group of cables from the different computers in each block.



(iv) The most economic way to connect it with a reasonable high speed would be to use radius 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 outdoor. 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.

2. Ravya Industries has set up its new centre at Kaka Nagar for its office and web based activities. The company compound has 4 buildings as shown in the diagram below:



Centre to centre distances between various building is as follows :

Harsh Building to Raj Building	50 m
Raj Building to Fazz Building	60 m
Fazz Building to Jazz Building	25m
Jazz Building to Harsh Building	170 m
Harsh Building to Fazz Building	125 m
Raj Building to Jazz Building	90m

Number of Computers in each of the buildings is as follows :

Harsh Building	15
Raj Building	150
Fazz Building	15
Jazz Building	25

(i) Suggest a cable layout of connections between the buildings.

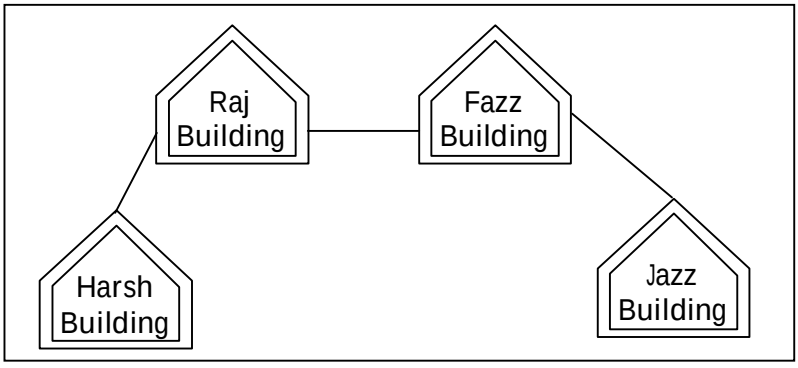
(ii) Suggest the most suitable place (i.e. building) to house the server of this organisation with a suitable reason.

(iii) Suggest the placement of the following devices with justification :

(a) Internet Connecting Device / Modem (b) Switch

(iv) The organisation is planning to link its sale counter situated in various parts of the same city, which type of network out LAN, MAN, or WAN will be formed?

Sol. (i)



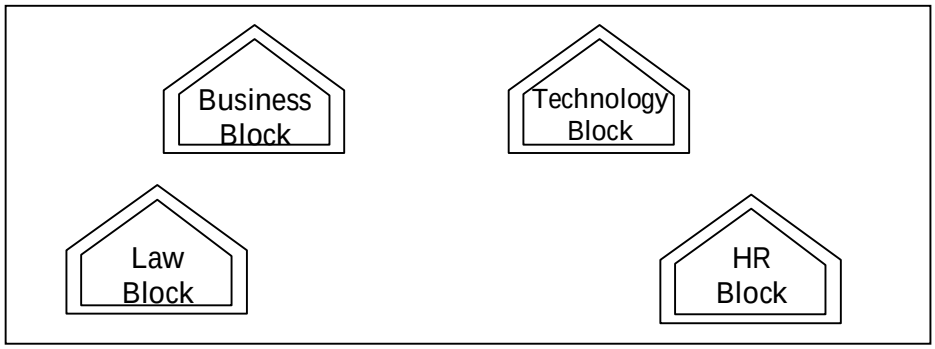
(ii) The most suitable place / block to house the server of this organisation would be Raj Building, as this block contains the maximum number of computers, thus decreasing the cabling cost for most of the computers as well as increasing the efficiency of the maximum computers in the network.

(iii) (a) Raj Building since it contains largest number of computers.

(b) In the suggest layout, a hub/switch each would be needed in all the buildings, to interconnect the group of cables from the different computers in each block.

(iv) The type of network that shall be formed to link the sale counters situated in various parts of the same city would be a MAN, because MAN (Metropolitan Area Networks) are the networks that link computer facilities within a city.

3. Quick Learn University is setting up its Academic blocks at Prayas Nagar and planning to set up a network. The university has 3 academic blocks and one Human Resource Centre as shown in the diagram below:



Centre to centre distances between various block/centre is as follows :

Law Block to Business Block	40 m
Law Block to Technology Block	80 m
Law Block to HR Centre	105 m
Business Block to Technology Block	30 m
Business Block to HR Centre	30 m
Technology Block to HR Centre	15 m

Number of Computers in each of the Block / Centre is as follows :

Law Block	15
Technology Block	40
HR Centre	115
Business Block	25

(i) Suggest the most suitable place (i.e., Block / Center) to install the server of this university with a suitable reason.

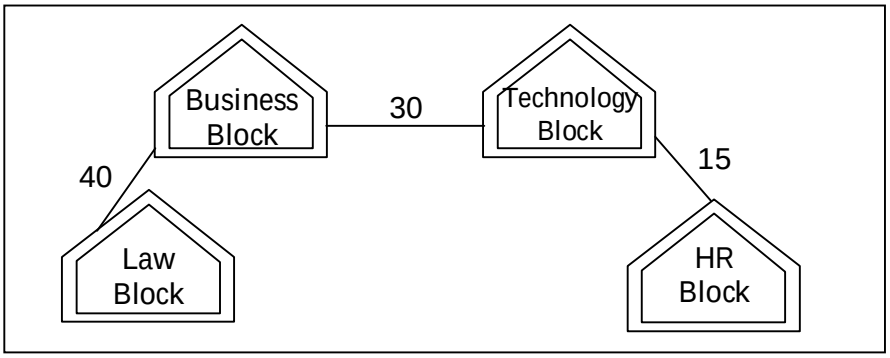
(ii) Suggest an ideal layout for connecting these blocks/centre for a wired connectivity.

(iii) Which device you will suggest to be placed/installed in each of these blocks/centre to efficiently connect all the computers with in these blocks/centre?

(iv) The university is planning to connect its admission office in the closest big city, which is more than 250 km form university, which type of network out of LAN, MAN or WAN will be formed ? Justify your answer.

Sol.: (i) HR Centre because it has the most number of computers.

(ii)

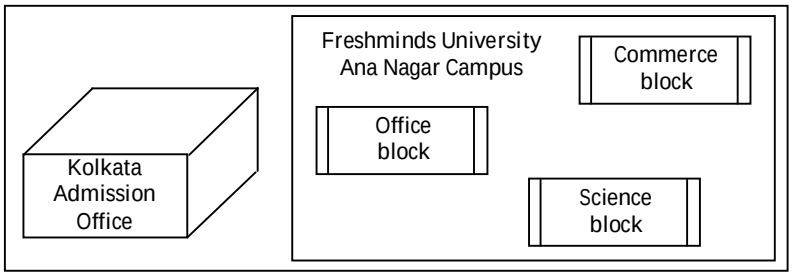


(iii) Switch

(iv) WAN because LAN and MAN cannot cover 250 km.

4. Freshminds University of India is starting its first campus in Ana Nagar of South India with its centre admission office in Kolkata. The university has 3 major blocks comprising of Office Block, Science Block and Commerice Block in the 5 km area Campus.

As a network expert, you need to suggest the network plane as per (i) to (iv) to the authorities keeping in mind the distance and other given parameters.



Expected Wire distances between various locations :

Office Block to Science Block	90 m
Office Block to Commerce Block	80 m
Science Block to Commerce Block	15 m
Kolkata Admission office to Ana Nagar Campus	2450 km

Expected number of Computers to be installed at various locations in the University are as follows :

Office Block	10
Science Block	140
Commerce Block	30
Kolkata Admission Block	8

(i) Suggest the authorities, the cable layout amongst various blocks inside university campus for connecting the blocks.

(ii) Suggest the most suitable place (i.e., block) to house the server of this university with a suitable reason.

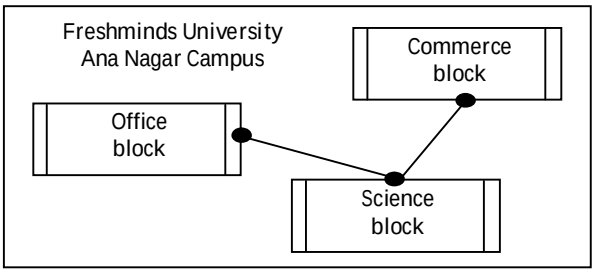
(iii) Suggest an efficient device from the following to be installed in each of the blocks to connect all the computers :

- MODEM
- SWITCH
- GATEWAY

(iv) Suggest the most suitable (very high speed) service to provide data connectivity between Admission Office looted in Kolkata and the campus located in Ana Nagar from the following options :

- Telephone line
- Fixed – Line Dial – up connection
- Co-axial Cable Network
- GSM
- Leased line
- Satellite Connection

Sol. (i)

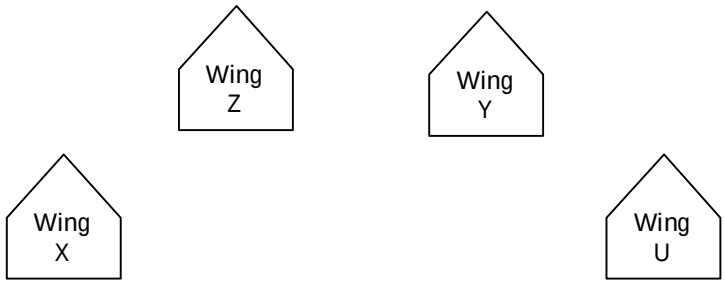


(ii) Science Block as it contains maximum number of computers.

(iii) SWITCH

(iv) Satellite Connection or Leased line

5. The Great Brain Organisation has set up the new Branch at Srinagar for its office and web based activities. It has 4 Wings of buildings as shown in the diagram :



Centre to centre distances between various block

Wing X to Wing Z	50 m
Wing Z to Wing Y	70 m
Wing Y to Wing X	125 m
Wing Y to Wing U	80 m
Wing X to Wing U	175 m
Wing Z to Wing U	90 m

Number of Computers

Wing X	50
Wing Z	30
Wing Y	150
Wing U	15

(i) Suggest the most suitable cable layout of connection between the Wings and topology.

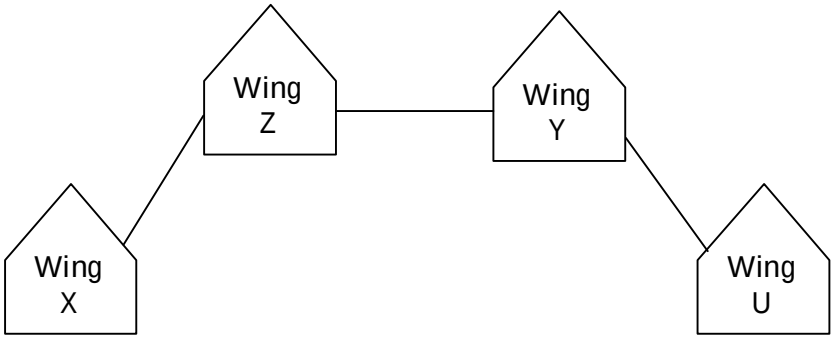
(ii) Suggest the most suitable paper (i.e., Wing) to house the server of this organisation

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

(a) Repeater (b) Hub/Switch

(iv) The organization is planning to link its head office situated in Delhi with the offices compromise on the speed of connectivity. Justify your answer.

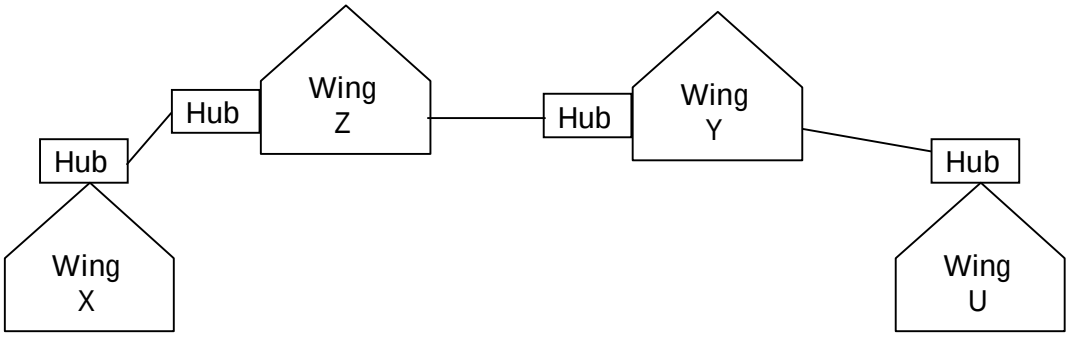
Sol. (i) **Bus Topology**



(ii) The most suitable place to house the server is **Wing Y** as it has the most number of computers thus cabling cost will be reduced and most traffic will be local.

(iii) (a) As per suggested layout separate repeaters need not be installed as each building/wing will be having a hub that acts a repeater.

(b) One hum per wing



(iv) An economic way of connecting is Dial-up or broadband as it can connect two computers at an economic rate through it provides lesser speed than other expensive methods.