

STUDENT SUPPORT MATERIAL
Class XII
COMPUTER SCIENCE



तत् त्वं पूषन् अपावृणु
केन्द्रीय विद्यालय संगठन

Session 2016-17

Kendriya Vidyalaya Sangathan
New Delhi



संतोष कुमार मल्ल, भा.प्र.से.
आयुक्त

Santosh Kumar Mall, I.A.S.
Commissioner



केन्द्रीय विद्यालय संगठन

KENDRIYA VIDYALAYA SANGATHAN

18, संस्थागत क्षेत्र, शहीद जीत सिंह मार्ग, नई दिल्ली-110016

दूरभाष : 91-11-26512579, फ़ैक्स : 91-11-26852680

18, Institutional Area, Shaheed Jeet Singh Marg, New Delhi-110016 (India)

Tel. : 91-11-26512579, Fax : 91-11-26852680

E-mail : commissioner@kvsedu.org, Website : www.kvsangathan.nic.in

A WORD TO MY DEAR STUDENTS

It gives me great pleasure in presenting the Students' Support Material to all KV students of class XII.

The material has been prepared keeping in mind your needs when you are preparing for final exams and wish to revise and practice questions or when you want to test your ability to complete the question paper in the time allotted or when you come across a question while studying that needs an immediate answer but going through the text book will take time or when you want to revise the complete concept or idea in just a minute or try your hand at a question from a previous CBSE Board exam paper or the Competitive exam to check your understanding of the chapter or unit you have just finished. This material will support you in any way you want to use it.

A team of dedicated and experienced teachers with expertise in their subjects has prepared this material after a lot of exercise. Care has been taken to include only those items that are relevant and are in addition to or in support of the text book. This material should not be taken as a substitute to the NCERT text book but it is designed to supplement it.

The Students' Support Material has all the important aspects required by you; a design of the question paper, syllabus, all the units/chapters or concepts in points, mind maps and information in tables for easy reference, sample test items from every chapter and question papers for practice along with previous years Board exam question papers.

I am sure that the Support Material will be used by both students and teachers and I am confident that the material will help you perform well in your exams.

Happy learning!

Santosh Kumar Mall
Commissioner, KVS

FOREWORD

The Students' Support Material is a product of an in-house academic exercise undertaken by our subject teachers under the supervision of subject expert at different levels to provide the students a comprehensive, yet concise, learning support tool for consolidation of your studies. It consists of lessons in capsule form, mind maps, concepts with flow charts, pictorial representation of chapters wherever possible, crossword puzzles, question bank of short and long answer type questions with previous years' CBSE question papers.

The material has been developed keeping in mind latest CBSE curriculum and question paper design. This material provides the students a valuable window on precise information and it covers all essential components that are required for effective revision of the subject.

In order to ensure uniformity in terms of content, design, standard and presentation of the material, it has been fine tuned at KVS Hqrs level.

I hope this material will prove to be a good tool for quick revision and will serve the purpose of enhancing students' confidence level to help them perform better. Planned study blended with hard work, good time management and sincerity will help the students reach the pinnacle of success.

Best of Luck.



U.N. Khaware
Additional Commissioner (Acad.)



STUDENT SUPPORT MATERIAL

ADVISORS

- Shri Santosh Kumar Mall, IAS, Commissioner, KVS (HQ), New Delhi
- Shri. U.N. Khaware, Addl. Commissioner (Academics), KVS (HQ)

CO-ORDINATION TEAM AT KVS (HQ)

- Dr. V. Vijayalakshmi, Joint Commissioner (Acad), KVS (HQ)
- Mr. P.V. Sai Ranga Rao, Deputy Commissioner (Acad), KVS (HQ)
- Ms. Aprajita, AEO (Acad), KVS (HQ)

CONTENT TEAM

- Mr. C. Mani, Deputy Commissioner, Gurgaon Region
- Mr. Nagendra Goel, Deputy Commissioner, Ranchi Region
- Mr. Lavendra Kumar Tyagi, PGT (Comp. Sc.), KV Sec. 8 Rohini
- Mr. Ranjeet Mamgain PGT (Comp. Sc.), KV No 2 Faridabad
- Ms. Neelima Singh, PGT (Comp. Sc.), KV Sec. 3 Rohini
- Ms. Kusum Kumari, PGT (Comp. Sc.), KV Hinoo (1st Shift) Ranchi
- Ms. Anjum Khatun, PGT (Comp. Sc.), KV CRPF Ranchi
- Mr. Nilmani Kumar, PGT (Comp. Sc.), KV Namkum Ranchi

REVIEW TEAM

- Mr. Bhupesh Bhatt, PGT (CS), KV AFS Rajokari
- Ms. Savita Kumari Chaurasia, PGT (CS), KV NMR Campus JNU

Typing Type-setting & Designing

M/s Vijaylakshmi Printing Works Pvt. Ltd.

B-117, Sector-5, Noida-201301, Ph.: 0120-2421977, 2422312

E-mail: vpwpl.1972@gmail.com

Table of Contents

Unit /Chapters	Page No.
Unit-I: Object Oriented Programming in C++	
Chapter 1: Revision Tour of Class XI	05-18
Chapter 2: Classes and Object	19-22
Chapter 3: Constructor and Destructor	23-24
Chapter 4: Inheritance	25-27
Chapter 5: Data File Handling	28-34
Unit-II : Data Structure	
Chapter 6: Arrays, Stacks, Queues And Linked List	35-46
Unit-III : DatabaseManagement Systems and SQL	
Chapter 7: DBMS & Structured Query Language	47-53
Unit-IV : Boolean Algebra	
Chapter 8: Boolean Algebra	54-59
Unit- V : Networking and Communication Technology	
Chapter 9: Networking and Communication Technology	60-66
Previous Year CBSE Question Papers	67-92

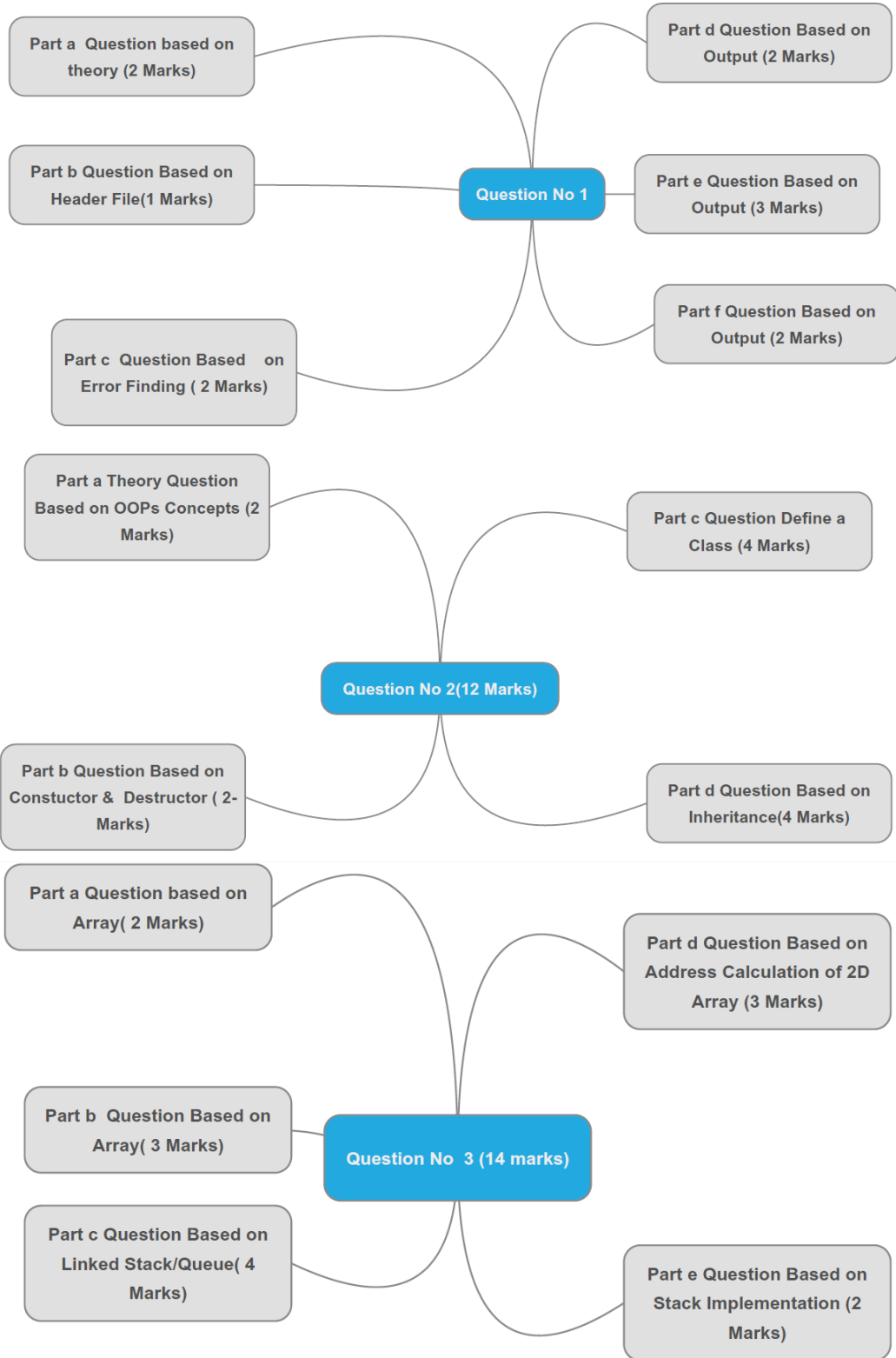
COURSE DESIGN

Unit	Topic	Marks
I	Object Oriented Programming in C++	30
II	Data Structure	14
III	DatabaseManagement Systems and SQL	8
IV	Boolean Algebra	8
V	Networking and Communication Technology	10
	TOTAL	70

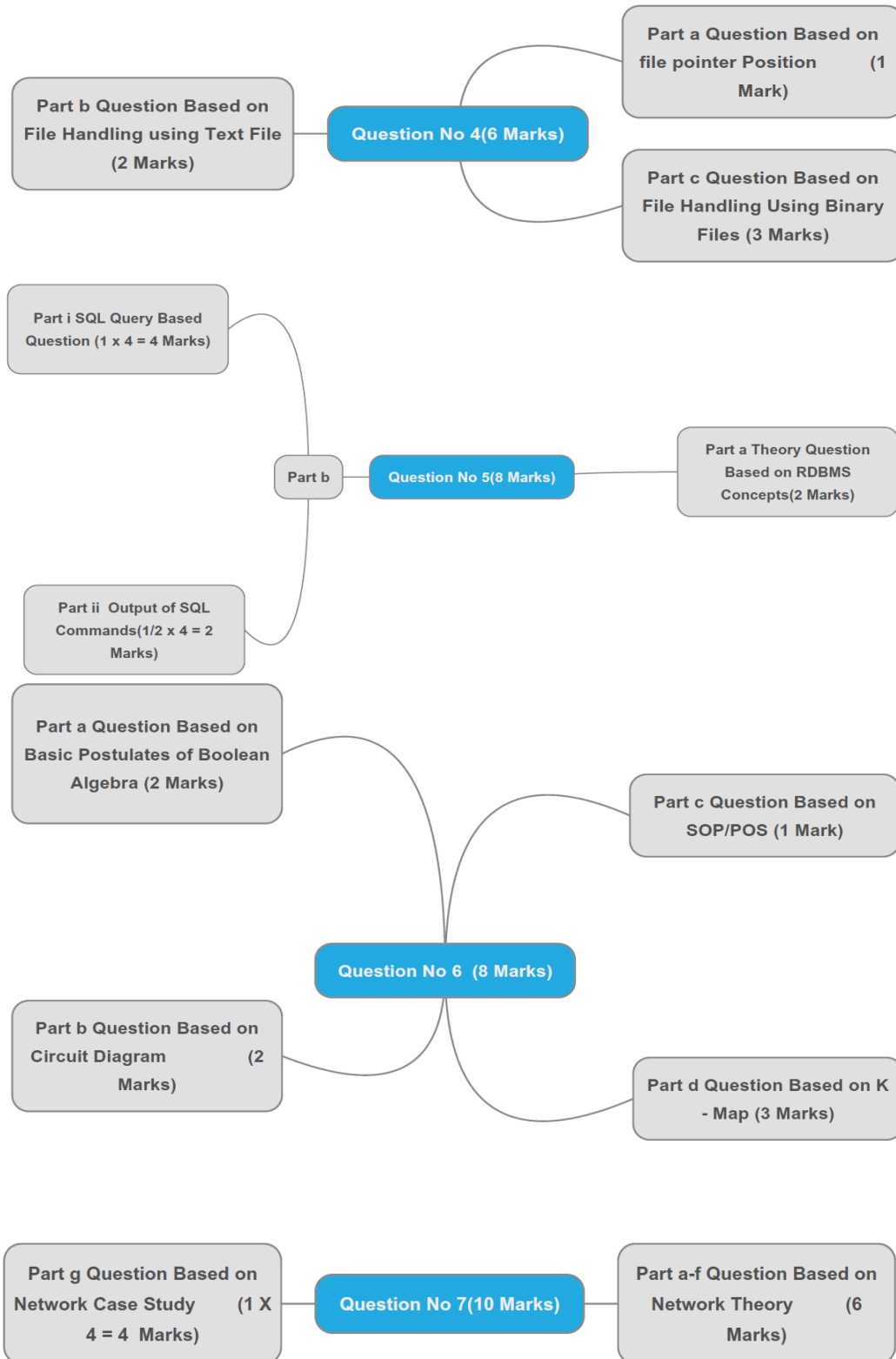
BLUE PRINT: CLASS XII
COMPUTER SCIENCE (083) Theory

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

Mindmap of CBSE Board Exam Question Paper (Question-wise)



Computer Science





Unit-I

Object Oriented Programming in C++

Revision Tour of Class XI

Chapter: 01

➤ **Keywords:** Keywords are the certain reserved words that convey a special meaning to the compiler. These are reserved for special purpose and must not be used as identifier name. eg for, if, else, this, do, etc.

➤ **Identifiers:** Identifiers are programmer defined names given to the various program elements such as variables, functions, arrays, objects, classes, etc.. It may contain digits, letters and underscore, and must begin with a letter or underscore. C++ is case sensitive as it treats upper and lower case letters differently. A keyword can not be used as an identifier. The following are some valid identifiers:

Pen time580 s2e2r3 _dos _HJI3_JK

➤ **Data Types in C++:** Data types are means to identify the types of data and associated operations of handling it. Data types in C++ are of two types:

1. Fundamental or Built-in data types: These data types are already known to compiler. These are the data types which are not composed of other data types. There are following fundamental data types in C++:

- (i) int data type (for integer)
- (ii) char data type (for characters)
- (iii) float data type (for floating point numbers)
- (iv) double data type

2. Derived and User defined data types : These data types are made up of fundamental data types : For example 1) Array 2) Function 3) Reference 4) Constant 5) Pointer 6) Class 7) Enum 8) Union 9) Structure

Data Type Modifiers: There are following four data type modifiers in C++ , which may be used to modify the fundamental data types to fit various situations more precisely:

- (i) signed
- (ii) unsigned
- (iii) long
- (iv) short

➤ **Variables:** A named memory location, whose contents can be changed with in program execution is known as variable. OR

A variable is an identifier that denotes a storage location, whose contents can be varied during program execution.

Declaration of Variables: Syntax for variable declaration is:

datatype variable_name1, variable_name2, variable_name3,..... ;

We can also initialize a variable at the time of declaration by using following syntax:

datatype variable_name = value;

When the initial value is given to the variable at the run time it is called dynamic initialization. e.g.

float avg;

avg = sum/count;

then above two statements can be combined in to one as follows:

float avg = sum/count;

➤ **Constant:** A named memory location, whose contents cannot be changed within program execution is known as constant. OR

A constant is an identifier that denotes a storage location, whose contents cannot be varied during program execution.

Syntax for constant declaration is:

const datatype constant_name = value ;

e.g., const float pi = 3.14f ;

➤ **Conditional operator (? :):**

The conditional operator (? :) is a ternary operator i.e., it requires three operands. The general form of conditional operator is:

expression1? expression2: expression3 ;

Where expression1 is a logical expression, which is either true or false.

If expression1 evaluates to true i.e., 1, then the value of the whole expression is the value of expression2, otherwise, the value of the whole expression is the value of expression3. For example

min = a < b ? a : b ;

Here if expression (a < b) is true then the value of a will be assigned to min otherwise the value of b will be assigned to min.

➤ **Type Conversion:** The process of converting one predefined data type into another is called type conversion.

C++ facilitates the type conversion in two forms:

(i) **Implicit type conversion:-** An implicit type conversion is a conversion performed by the compiler without programmer's intervention. An implicit conversion is applied generally whenever different data types are intermixed in an expression. The C++ compiler converts all operands up to the data type of the largest data type's operand, which is called type promotion.

(ii) **Explicit type conversion :-** An explicit type conversion is user-defined that forces an expression to be of specific data type.

Type Casting:- The explicit conversion of an operand to a specific type is called type casting.

Type Casting Operator - (type) :- Type casting operators allow you to convert a data item of a given type to another data type. To do so, the expression or identifier must be preceded by the name of the desired data type, enclosed in parentheses. i. e.,

(data type) expression

Where data type is a valid C++ data type to which the conversion is to be done. For example, to make sure that the expression (x+y/2) evaluates to type float, write it as:

(float) (x+y/2)

➤ **Some important Syntax in C++:**

1. if Statement

```

if ( < conditional expression > )
{
< statement-1 or block-1>;
// statements to be executed when conditional expression is true.
}
[ else
{
< statement-2 or block-2>;
// statements to be executed when conditional expression is false.
} ]

```

2. The if-else-if ladder :

```

if ( <condition -1> )
    statement-1; // do something if condition-1 is satisfied (True)
else if ( <condition - 2 > )
    statement-3 ; // do something if condition -2 is satisfied (True)
else if ( <condition - 3 > )
    statement-3 ; // do something if condition- 3 is satisfied (True)
:

```

```

: // many more n-1 else - if ladder may come
:
else if( < condition – n >)
    statement-n ;      // do something if condition – n is satisfied (True)
[   else
    statement-m ;   ] // at last do here something when none of the
                    // above conditions gets satisfied (True)
}

```

<> in syntax is known as a place holder, it is not a part of syntax, do not type it while writing program. It only signifies that anything being kept there varies from program to program.

[] is also not a part of syntax , it is used to mark optional part of syntax i.e. all part of syntax between [] is optional.

3. switch Statement :-

```

switch (expression/variable)
{
    case value_1: statement -1;
                break;
    case value_2: statement -2;
                break;
    :
    :
    case value_n: statement -n;
                break;
    [ default: statement -m; ]
}

```

4. The for Loop:

```

for(initialization_expression(s); loop_Condition; update_expression (s))
{
    Body of loop
}

```

5. while Loop:

```

while (loop_condition)
{
    Loop_body
}

```

6. do-while loop:

```

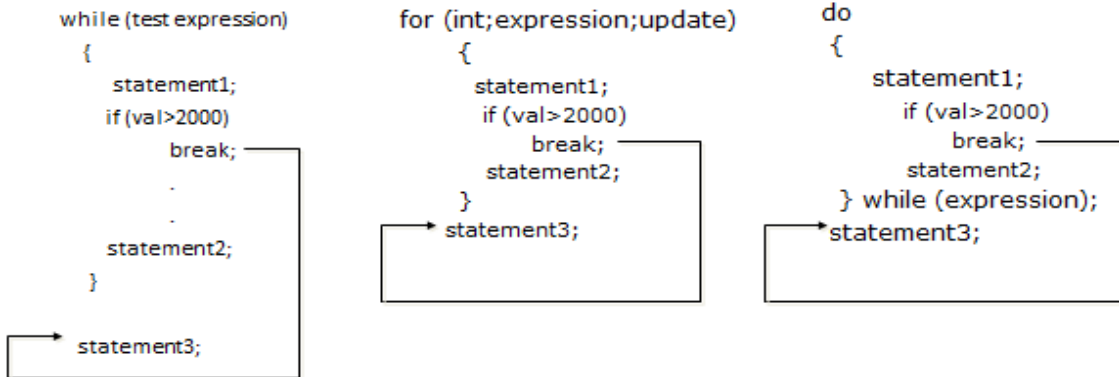
do
{
    Loop_body
}while (loop_condition);

```

➤ **break Statement :-**The break statement enables a program to skip over part of the code. A break statement terminates the smallest enclosing while, do-while, for or switch statement. Execution resumes at the statement immediately following the body of the terminated statement. The following figure explains the working of break statement:

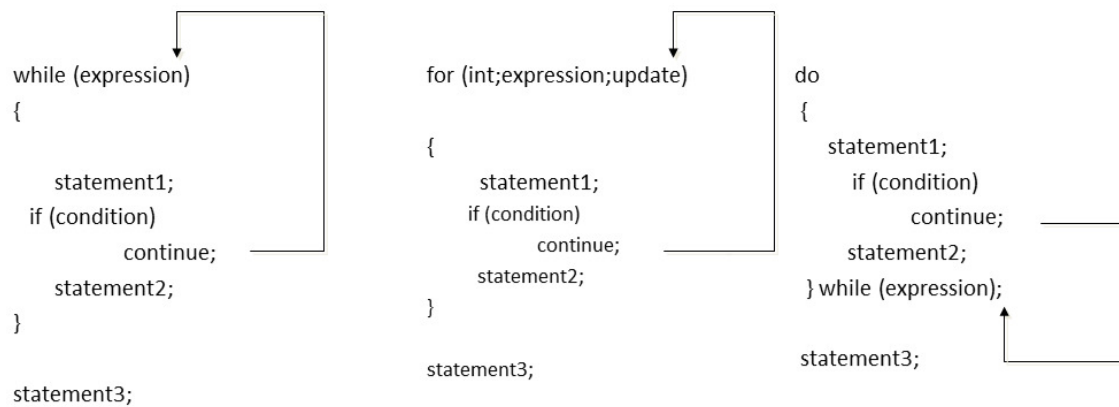
Computer Science

The Working of Break Statement



➤ **continue Statement:-** The continue is another jump statement like the break statement as both the statements skip over a part of the code. But the continue statement is somewhat different from break. Instead of forcing termination, it forces the next iteration of the loop to take place, skipping any code in between. The following figure explains the working of continue statement:

The Working of Continue Statement



➤ **Functions :-** Function is a named group of programming statements which perform a specific task and return a value.

1. Built-in Functions (Library Functions) :- The functions, which are already defined in C++ Library (in any header files) and a user can directly use these function without giving their definition is known as built-in or library functions. e.g., sqrt(), toupper(), isdigit() etc.

Following are some important Header files and useful functions within them :

stdio.h (standard I/O function)	gets(), puts()
cctype.h (character type function)	isalnum(), isalpha(), isdigit(), islower(), isupper(), tolower(), toupper()
string.h (string related function)	strcpy(), strcat(), strlen(), strcmp(), strcmpi(), strrev(),strupr(), strlwr()
math.h (mathematical function)	fabs(), pow(), sqrt(), sin(), cos(), abs()
stdlib.h	randomize(), random()



randomize() : This function provides the seed value and an algorithm to help random() function in generating random numbers. The seed value may be taken from current system's time.

random(<int>) : This function accepts an integer parameter say x and then generates a random value between 0 to x-1.

for example : random(7) will generate numbers between 0 to 6.

To generate random numbers between a lower and upper limit we can use following formula

$$\text{random}(U - L + 1) + L$$

where U and L are the Upper limit and Lower limit values between which we want to find out random values.

For example : If we want to find random numbers between 10 to 100 then we have to write code as:

```
random(100 - 10 + 1) + 10 ; // generates random number between 10 to 100
```

2. User-defined function :- The functions which are defined by user for a specific purpose is known as user-defined function. For using a user-defined function it is required, first define it and then using.

Declaration of user-defined Function:

```
Return_type function_name(List of formal parameters)
{
    Body of the function
}
```

Calling a Function:- When a function is called then a list of actual parameters is supplied that should match with formal parameter list in number, type and order of arguments.

Syntax for calling a function is:

```
function_name ( list of actual parameters );
```

e.g.,

```
#include <iostream.h>
int addition (int a, int b)
{ int r;
  r=a+b;
  return (r); }
void main ( )
{ int z ;
  z = addition (5,3);
  cout<< "The result is " << z;
  }
The result is 8
```

```
int addition (int a, int b)
```



```
z = addition ( 5 , 3 );
```

```
int addition (int a, int b)
```



```
z = addition ( 5 , 3 );
```

Call by Value (Passing by value) :- The call by value method of passing arguments to a function copies the value of actual parameters into the formal parameters, that is, the function creates its own copy of argument values and then use them, hence any change made in the parameters in function will not reflect on actual parameters. The above given program is an example of call by value.

Call by Reference (Passing by Reference) :- The call by reference method uses a different mechanism. In place of passing value to the function being called, a reference to the original variable is passed. This means that in call by reference method, the called function does not create its own copy of original values, rather, it refers to the original values only by different names i.e., reference. Thus the called function works on the original data and any changes are reflected to the original values.

// passing parameters by reference

```
#include <iostream.h>
void duplicate (int& a, int& b, int& c)
```



```

{
    a*=2;
    b*=2;
    c*=2;
}

void main ()
{
    int x=1, y=3, z=7;
    duplicate (x, y, z);
    cout <<"x="<< x <<" , y="<< y <<" , z="<< z;
}
    
```

void duplicate (int& a,int& b,int& c)

↕ x ↕ y ↕ z

duplicate (x , y , z);

output :x=2, y=6, z=14

The ampersand (&)operator specifies that their corresponding arguments are to be passed *by reference* instead of *by value*.

Constant Arguments:-In C++ the value of constant argument cannot be changed by the function.

To make an argument constant to a function , we can use the keyword `const` as shown below:

```
int myFunction( const int x , const int b );
```

The qualifier `const` tell the compiler that the function should not modify the argument. The compiler will generate an error when this condition is violated.

Default Arguments :- C++ allows us to assign default value(s) to a function’s parameter(s) which is useful in case a matching argument is not passed in the function call statement. The default values are specified at the time of function definition from right most parameter to left ones. e.g.,

```
float interest ( float principal, int time, float rate = 0.70f)
```

Here if we call this function as:

```
si_int= interest(5600,4);
```

then rate =0.7 will be used in function.

Formal Parameters:- The parameters that appear in function definition are formal parameters.

Actual Parameters :- The parameters that appears in a function call statement are actual parameters.

Functions with no return type (The use of void):- In this case we should use the void type specifier for the function. This is a special specifier that indicates absence of type.

The return Statement :- The execution of return statement, it immediately exit from the function and control passes back to the calling function (or, in case of the main(), transfer control back to the operating system). The return statement also returns a value to the calling function. The syntax of return statement is:

```
return ( value);
```

➤ **Scope of Identifier :-**The part of program in which an identifier can be accessed is known as scope of that identifier. There are four kinds of scopes in C++

- (i) **Local Scope :-** An identifier declare in a block ({ }) is local to that block and can be used only in it.
- (ii) **Function Scope :-** The identifier declare in the outermost block of a function or in its argument list, have function scope.



- (iii) **File Scope (Global Scope)** :- An identifier has file scope or global scope if it is declared outside all blocks i.e., it can be used in all blocks and functions.
- (iv) **Class Scope** :- A name of the class member has class scope and is local to its class.
- **Lifetime** :The time interval for which a particular identifier or data value lives in the memory is called Lifetime of the identifier or data value.

➤ **Arrays:**

Declaration of One-Dimensional Array:-

Data_type Array_name[size];

Working with One Dimensional Array:-

General form of for loop for Reading elements of array (1-D)	Generally processing part may be include within the loop of reading or printing, otherwise a same type separate loop may be used for processing	General form of for loop for printing elements of array (1-D)
<pre>for (int i=0; i< size; i++) { cout<<"Enter Array Element "<<i+1; cin>>Array_Name[i]; }</pre>		<pre>for (int i=0; i< size; i++) { cout<<Array_Name[i]<< " , "; }</pre>

Declaration of 2-D array:-

Data_type Array_name [R][C] ;

Where R represent number of rows and C represent number of columns in array.

Working With Two-Dimensional Array:-

General form of for loop for Reading elements of 2-D array	Generally processing part may be include within the loop of reading or printing, otherwise a same type separate nested loop may be used for processing	General form of for loop for printing elements of 2-D array
<pre>for (int i=0; i< R; i++) { cout<<"Enter Row "<<i+1; for (int j=0; j<C ; j++) cin>>Array_Name[i][j]; }</pre>		<pre>for (int i=0; i< R; i++) { for (int j=0; j<C ; j++) cout<<Array_Name[i][j] <<'t'; cout<<'n'; }</pre>

Where R represent number of rows and C represent number of columns in array.

➤ **Defining Structure :-**

```
struct< Name of Structure >
{
  <datatype>< data-member 1>;
  <datatype>< data-member 2>;
  <datatype>< data-member 3>;
  ...
  ...
  <datatype>< data-member n>;
};
```

Declaring Structure Variable :-

```
struct< Name of Structure >
{
  <datatype>< data-member 1>;
  <datatype>< data-member 2>;
```



```
<datatype>< data-member 3>;  
    ...  
    ...  
<datatype>< data-member n>;  
} var1, var2,....., varn ;
```

We can declare the structure type variables separately (after defining of structure) using following syntax:

```
Structure_name var1, var2, ..... ....., var_n;
```

Accessing Structure Elements :- To access structure element , dot operator is used. It is denoted by (.). The general form of accessing structure element is :
Structure_Variable_Name.element_name

➤ **Pointer:-** Pointer is a variable that holds a memory address of another variable of same type.

Declaration and Initialization of Pointers :

Syntax :

```
Datatype *variable_name;
```

```
e.g., int *p;          float *p1;          char *c;
```

Two special unary operator * and & are used with pointers. The & is a unary operator that returns the memory address of its operand.

```
e.g., int a = 10;    int *p;          p = &a;
```

Pointer arithmetic: Two arithmetic operations, addition and subtraction, may be performed on pointers. When you add 1 to a pointer, you are actually adding the size of whatever the pointer is pointing at. That is, each time a pointer is incremented by 1, it points to the memory location of the next element of its base type.

```
e.g. int *p;          p++;
```

If current address of p is 1000, then p++ statement will increase p to 1002, not 1001.

Adding 1 to a pointer actually adds the size of pointer's base type.

Base address : A pointer holds the address of the very first memory location of array where it is pointing to. The address of the first memory location of array is known as BASE ADDRESS.

Dynamic Allocation Operators : C++ dynamic allocation operators allocate memory from the free store/heap/pool, the pool of unallocated heap memory provided to the program. C++ defines two operators **new** and **delete** that perform the task of allocating and freeing memory during runtime.

Pointers and Arrays : C++ treats the name of an array as constant pointer which contains base address i.e address of first memory location of array.

➤ **typedef :-** The typedef keyword allows to create alias(alternate name) for data types. the syntax is:

```
typedef existing_data_type new_name ;
```

```
e.g. typedef int num;
```

➤ **#define Preprocessor Directive:** The # define directive creates symbolic constant, constants that are represent as macros.

Macros: Macros are preprocessor directive created using # define that serve as symbolic constants. They are created to simplify and reduce the amount of repetitive coding

e.g.1

```
#define PI 3.14
```

Here PI is defined as a macro. It will replace 3.14 in place of PI throughout the program.

e.g. 2

```
#define max (a, b) a>b? a: b
```

Defines the macro max, taking two arguments a and b. This macro may be called like any

function. Therefore, after preprocessing

$A = \max(x, y)$;

Becomes $A = x > y ? x : y$;

- **Function Overloading:** Function overloading is the process of defining and using functions with same name having different argument list and/or different return types. These functions are differentiated during the calling process by the number, order and types of arguments passed to these functions.

Example:

`int Add (int ,int) ;`

`double Add (double ,double) ;`

`float Add (int ,float) ;`

Short Answer Type Questions (2-Marks)

1. Define Macro with suitable example.
2. Explain in brief the purpose of function prototype with the help of a suitable example.
3. What is the role of typedef? Can it be used to create new data type?
4. What is the difference between Object Oriented Programming and Procedural Programming?
5. What is the difference between Global Variable and Local Variable? Also, give a suitable C++ code to illustrate both.
6. Differentiate between ordinary function and member functions in C++. Explain with an example.
7. What is the difference between call by reference and call by value with respect to memory allocation? Give a suitable example to illustrate using C++ code.
8. What is the difference between actual and formal parameter ? Give a suitable example to illustrate using a C++ code.
9. Differentiate between a Logical Error and Syntax Error. Also give suitable examples of each in C++.
10. Find the correct identifiers out of the following, which can be used for naming variable, constants or functions in a C++ program :
While, for, Float, new, 2ndName, A%B, Amount2, _Counter
11. Out of the following, find those identifiers, which cannot be used for naming Variable, Constants or Functions in a C++ program :
_Cost, Price*Qty, float, Switch, Address One, Delete, Number12, do
12. Find the correct identifiers out of the following, which can be used for naming Variable, Constants or Functions in a C++ program :
For, while, INT, NeW, delete, 1stName, Add+Subtract, name1

Very Short Answer Type Questions (1-Mark Based on Header Files)

1. Which C++ header file (s) will be included to run /execute the following C++ code?

```
void main()
{ int Last =26.5698742658;
  cout<<setw(5)<<setprecision(9)<<Last; }
```

Ans: iostream.h, iomanip.h

2. Name the header files that shall be needed for successful compilation of the following C++ code :

```
void main()
{ char str[20],str[20];
```



Computer Science

```
gets(str);
strcpy(str1,str);
strev(str);
puts(str);
puts(str1);    }
```

3. Write the names of the header files to which the following belong:
(i) strcmp() (ii) fabs()
4. Write the names of the header files to which the following belong:
(i) frexp() (ii) isalnum()

Short Answer Type Questions (2-Marks Error Finding)

1. Rewrite the following program after removing any syntactical errors. Underline each correction made.

```
#include<iostream.h>
void main( )
int A[10];
A=[3,2,5,4,7,9,10];
for( p = 0; p<=6; p++)
{ if(A[p]%2=0)
int S = S+A[p]; }
cout<<S;
}
```

```
Ans :- #include<iostream.h>
void main( )
{int A[10] = {3,2,5,4,7,9,10};
int S = 0,p;
for(p = 0; p<=6; p++)
{ if(A[p]%2==0)
S = S+A[p]; }
cout<<S;}
```

2. Deepa has just started working as a programmer in STAR SOFTWARE company. In the company she has got her first assignment to be done using a C++ function to find the smallest number out of a given set of numbers stored in a one-dimensional array. But she has committed some logical mistakes while writing the code and is not getting the desired result. Rewrite the correct code underlining the corrections done. Do not add any additional statements in the corrected code

```
int find(int a[],int n)
{ int s=a[0];
for(int x=1;x<n;x++)
if(a[x]>s)
a[x]=s;
return(s); }
```

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

```
#include [iostream.h]
class PAYITNOW
{ int Charge;
PUBLIC:
void Raise(){cin>>Charge;}
void Show{cout<<Charge;}
};
void main()
{
PAYITNOW P;
P.Raise();
Show();
}
```

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

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

Short Answer Type Questions (2-Marks Finding Output)

<p>2. Find the output of the following C++ program:</p> <pre>#include<iostream.h> void repch(char s[]) { for (int i=0;s[i]!='\0';i++) { if((i%2)!=0) &&(s[i]!=s[i+1]) { s[i]='@'; } else if (s[i]==s[i+1]) { s[i+1]='!'; } } } void main() { char str[]="SUCCESS"; cout<<"Original String"<<str; repch(str); cout<<"Changed String"<<str;} Ans: Original String SUCCESS Changed String S@C@ES!</pre>	<p>1. Find output of the following program segment :</p> <pre>#include<iostream.h> #include<ctype.h> void Mycode(char Msg[],char CH) { for(int cnt=0;Msg[cnt]!='\0';cnt++) { if(Msg[cnt]>='B'&& Msg[cnt]<='G') Msg[cnt]=tolower(Msg[cnt]); else if(Msg[cnt]=='N' Msg[cnt]=='n' Msg[cnt]==' ') Msg[cnt]=CH; else if(cnt%2==0) Msg[cnt]=toupper(Msg[cnt]); else Msg[cnt]=Msg[cnt-1]; } } void main() { char MyText[]="Input Raw"; Mycode(MyText,'@'); cout<<"NEW TEXT:"<<MyText<<endl; }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>3. Find the output of the following program:</p> <pre>#include <iostream.h> #include <ctype.h> void Encrypt(char T[]) { for (int i=0;T[i]!='\0';i+=2) if (T[i]=='A' T[i]=='E') T[i]='#'; else if (islower(T[i])) T[i]=toupper(T[i]); else T[i]='@'; } void main() { char Text[]="SaVE EaRtH"; //The two words in the string Text //are separated by single space Encrypt(Text); cout<<Text<<endl; }</pre>	<p>4. Find the output of the following program:</p> <pre>#include <iostream.h> struct Game { char Magic[20]; int Score; }; void main() { Game M={"Tiger",500}; char *Choice; Choice=M.Magic; Choice[4]='P'; Choice[2]='L'; M.Score+=50; cout<<M.Magic<<M.Score<<endl; Game N=M; N.Magic[0]='A';N.Magic[3]='J'; N.Score-=120; cout<<N.Magic<<N.Score<<endl; }</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Application Based Questions (3 Marks Finding Output)

<p>1. Find the output of the following :</p> <pre>#include<iostream.h> void switchover(int A[],int N, int split) { for(int K = 0; K<N; K++) if(K<split) A[K] += K; else A[K]*= K; } void display(int A[],int N) { for(int K = 0; K<N; K++) (K%2== 0) ?cout<<A[K]<<"%" : cout<<A[K]<<endl; } void main() { int H[] = {30,40,50,20,10,5}; switchover(H,6,3); display(H,6); }</pre> <p>Ans : 30%41 52%60 40%25</p>	<p>2. Find the output of the following program :</p> <pre>#include<iostream.h> void in(int x,int y, int &z) { x+=y; y- -; z*=(x-y); } void out(int z,int y, int &x) { x*=y; y++; z/=(x+y); } void main() { int a=20, b=30, c=10; out(a,c,b); cout<<a<<"#"<<b<<"#"<<c<<"#"<<endl; in(b,c,a); cout<<a<<"@"<<b<<"@"<<c<<"@"<<endl; out(a,b,c); cout<<a<<"\$"<<b<<"\$"<<c<<"\$"<<endl; }</pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>3. Find the output of the following program:</p> <pre>#include <iostream.h> struct PLAY { int Score, Bonus;}; void Calculate(PLAY &P, int N=10) { P.Score++;P.Bonus+=N; } void main() { PLAY PL={10,15}; Calculate(PL,5); cout<<PL.Score<<"."<<PL.Bonus<<endl; Calculate(PL); cout<<PL.Score<<"."<<PL.Bonus<<endl; Calculate(PL,15); cout<<PL.Score<<"."<<PL.Bonus<<endl; }</pre>	<p>4. Find the output of the following program:</p> <pre>#include <iostream.h> void Changethecontent(int Arr[], int Count) { for (int C=1;C<Count;C++) Arr[C-1]+=Arr[C]; } void main() { int A[]={3,4,5},B[]={10,20,30,40},C[]={900,1200}; Changethecontent(A,3); Changethecontent(B,4); Changethecontent(C,2); for (int L=0;L<3;L++) cout<<A[L]<<'#'; cout<<endl; for (L=0;L<4;L++) cout<<B[L] <<'#'; cout<<endl; for (L=0;L<2;L++) cout<<C[L] <<'#'; } }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

5 Find the output of the following program:

```
#include <iostream.h>
struct GAME
{ int Score, Bonus;};
void Play(GAME &g, int N=10)
{ g.Score++;g.Bonus+=N; }
void main()
{ GAME G={ 110,50};
  Play(G,10);
  cout<<G.Score<<"."<<G.Bonus<<endl;
  Play(G);
  cout<<G.Score<<"."<<G.Bonus<<endl;
  Play(G,15);
  cout<<G.Score<<"."<<G.Bonus<<endl; }
```

Application Based Questions(2 Marks Based on random function)

1. Observe the following C++ code and find out , which out of the given options i) to iv) are the expected correct output. Also assign the maximum and minimum value that can be assigned to the variable 'Go'.

```
void main()
{ int X [4] = { 100,75,10,125};
  int Go = random(2)+2;
  for (int i = Go; i < 4; i++)
  cout<<X[i]<<"$$";
}
```

Ans :
 (iv) is the correct option.
 Minimum value of Go = 2
 Maximum value of Go = 3



Computer Science

- (i) 100\$75 (ii) 75\$10\$125\$\$ (iii) 75\$10\$\$ (iv) 10\$125\$

2. In the following program, if the value of N given by the user is 15, what maximum and minimum values the program could possibly display?

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

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

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

4. Read the following C++ code carefully and find out, which out of the given options (i) to (iv) are the expected correct output(s) of it. Also, write the maximum and minimum value that can be assigned to the variable Taker used in the code :

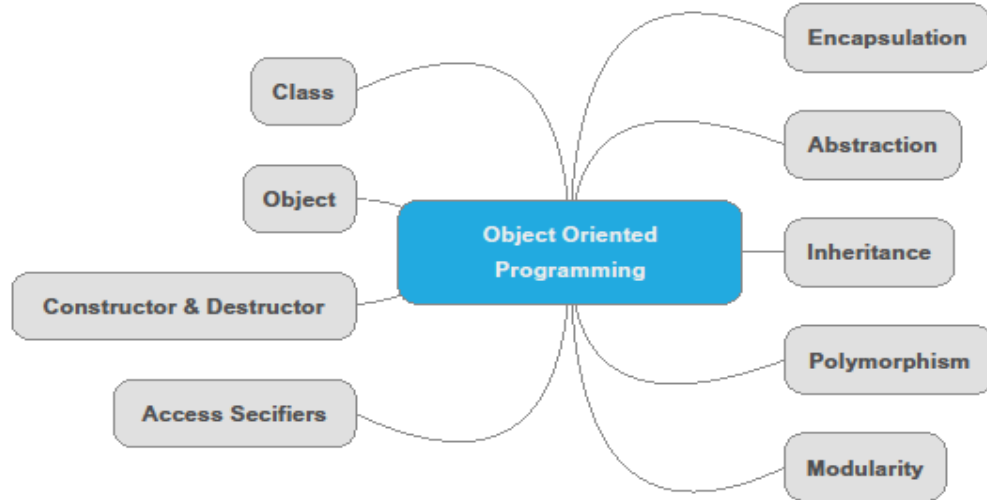
```
void main()
{
    int GuessMe[4]={ 100,50,200,20};
    int Taker=random(2)+2;
    for (int Chance=0;Chance<Taker;Chance++)
    cout<<GuessMe[Chance]<<"#"; }
}
```

- (i) 100# (ii) 50#200# (iii) 100#50#200# (iv) 100#50

Unit-I
Object Oriented Programming in C++

OOP, Classes And Objects

Chapter: 02



Class :- A class is collection of data (data member) and functions (member functions or methods) working on the data. It can be seen as a blue print for the object. No memory is allocated when a class is created. Memory is allocated only when an object is created.

Object :- An Object is an instance of the class means have all the properties defined in the class.

Data member:- The data variables declared within the class.

Member functions :- Member functions are the methods which are declared/defined inside the class and operate upon the data member.

Data Abstraction: - Data abstraction represents essential features without including background details.

Data Encapsulation:- Binds the data and its functions into a single unit called class.

Data hiding:-Hides internal object details (data members). Data hiding ensures exclusive data access to class members and protects object integrity by preventing accidental or intended changes.

Inheritance: Inheritance is the process of forming a new class from an existing class or base class.

Base Class :- The class from which methods and data members are derived to new class is knows as base class. The base class is also known as parent class or super class.

Derived Class:- The class that is deriving data and methods from base class is called derive class. Derived class is also known as a child class or sub class.

Polymorphism:-Poly means many and **morphs** mean form (Multiple Forms). Refers to the ability of processing of data in more than one form.

Access specifier :-private, protected, public (default access specifier is private)

Accessibility of private, protected and public members

Accessibility	Private	Protected	Public
Through member functions	Yes	Yes	Yes
Through object of the class	No	No	Yes
Through derived class	No	Yes	Yes

Computer Science

Syntax of a class	Example
<pre>class <class_name> { private: declaration of data member; declaration/definition member function; protected: declaration of data member; declaration/definition member function public: declaration of data member; declaration/definition member function };</pre>	<pre>class student { private: char name[30]; int age; int marks; protected: char grade; public: void getdata(); void showdata(); };</pre>

Referencing class members:- All the data members of the class are directly accessible to the member function of that class. They don't need any object name to be prefixed before it but from outside the class any reference to the data member is done with the dot (.) operator.

syntax for creating an object:

<class_name><Object_name>;

Example:

student s1;

Accessing members from object of the class:-A data member and member function declared under public access specifier can be accessed by the objects directly.

objectname.member;

e.g.

s1.getdata();

s1.showdata();

Defining class methods/Member functions. Member functions of the class can be defined in the following two ways

(a) Inside the class definition (inline function)

In this method, the function is defined within the class body and are treated as inline by default.

(b) Outside the class definition.

In this way function prototype is declared within class body and function is defined outside the class with the help of Scope Resolution operator (::).

Syntax for defining a member function outside the class definition.	Example for defining a member function outside the class definition.
<pre><return type><class name> :: <function name>(parameter list) { body of the function }</pre>	<pre>void student::showdata() { cout<<"\n Name "<<name; cout<<"\n Age "<<age; cout<<," \n Marks"<<marks; }</pre>

Example: Defining member function within class body.	Example: Defining member function outside class body.
<pre> class STUDENT { int rollno; char Name[20] ; char Address[30] ; char PhoneNo[15] ; public : void enter() { cout<<"\n Enter Rollno name address and phone no. "; cin>>rollno; cin.getline(Name,20); cin.getline(Address,30); cin.getline(PhoneNo,15); } void display() { cout<<"information of student is"; cout<<rollno<<Name<<Address<<PhoneNo; } } ; </pre>	<pre> class STUDENT { int rollno; char Name[20] ; char Address[30] ; char PhoneNo[15] ; public : void enter(); void display(); } ; void STUDENT :: enter() { cin>>rollno; cin.getline(Name,20); cin.getline(Address,30); cin.getline(Phoneno,15); } void STUDENT :: display() { cout<<"information of student is"; cout<<rollno<<Name<<Address<<Phoneno; } </pre>



Computer Science

Short Answer Type Questions (2 Marks)

- Q1. What do you understand by class and object in C++?
Q2. What all members of a class are accessible to outside the class body?
Q3. What do you understand by data abstraction?
Q4. What do you understand by data hiding?
Q5. What do you understand by Encapsulation?
Q6. What is polymorphism?
Q7. What do you understand by member function? How does a member function differ from an ordinary function?
Q8. How does a class implement Data Abstraction and Encapsulation?

Long Answer Type Questions (4 Marks)

Q1. Define a class train with following members.

Private members

Trainno - type int

Destination -type String

Distance - type float

Fuel - type float

A member function calfuel() to calculate and assign value of fuel as per the following criteria:

Distance	Fuel
≤ 1500	250
> 1500 and ≤ 3000	1000
> 3000	2500

Public member:

1. feedinfo(): to input train no, destination, distance and invoke calfuel() function to assign value of fuel.
2. showinfo(): to display all the details for a train.

Q2. Define a class employee with following specifications:

Private members:

empno integer type.

ename 20 characters long String.

basic,hra,da float type.

netpay float type.

ctotal() A function to calculate the total basic.

Public member:

read_data() A function to read empno, ename, basic, hra, da and call ctotal () to calculate total.

display_data() A function to display all the data members on the screen.



Unit-I Object Oriented Programming in C++

Constructors and Destructors

Chapter: 03

Constructor:- A constructor is a special member function with the same name as its class name and is used to initialize the data members of the class. Constructor is invoked automatically when an object of a class is created. Constructor do not return any value not even void. Constructor must be defined in public section.

Types of Constructors

1. **Default Constructor (No argument constructor):-** A default constructor accepts no parameters. When no constructor is defined in the class, compiler provides the default constructor.
2. **Parameterized Constructor (Overloaded Constructor):-** Parameterized constructor accepts parameters and is used to assign these parameters to the data members of the class. There may be many definitions of the parameterized constructor depending upon the type and number of parameters passed to the constructor and so it is also called overloaded constructor.
3. **Copy Constructor:-** A constructor that accepts a reference to an instance of its own class as an argument is called as Copy Constructor. A copy constructor is used to create new object with the similar values of existing object. A copy constructor is invoked when one object is defined and initialized with another object of the same class.

Syntax for declaration of copy constructor:-

classname(classname &obj) ;

for example:- Student(Student &s)

Example of three different types of constructors. (default, parameterized, copy).	
<pre>#include<iostream.h> class student { int rollno; float percentage; public: student() // default constructor { rollno=0; percentage=0.0; } //parameterized constructor student(int rno,float p) { rollno=rno; percentage=p; } student(student &s) // copy constructor { rollno=s.rollno; percentage=s.percentage; }</pre>	<pre>void display() { cout<<"RNo. "<<rollno; cout<<"\n per "<<percentage; } }; void main() { //call for the default constructor student s; //call for the parametrized constructor student s1(5,88.5); //call for the copy constructor student s2=s1; s1.display(); s2.display(); }</pre>

Note 1 : When parameterized constructor is defined one must define the default constructor also, otherwise error may occur when a call to default constructor is made.



Computer Science

Note 2: When multiple constructors are defined for a class it is also known as constructor overloading.

Destructor: A function having same name as of class preceded with a tilde(~) sign, which is automatically called when scope of object is over. It is used to deallocate the memory.

Note: The order of destructor invocation is reverse of constructor invocation.

Short Answer Type Questions (2 Marks)

Q1. What do you understand by constructor and destructor?

Q2. What are different types of constructors?

Q3. What do you understand by Default constructor? What is its role?

Q4. Answer the questions (i) and (ii) after going through the following class-

```
class Race
{ int CarNo,Track;
public:
    Race(); //function 1
    Race(int CN); //function 2
    Race(Race &R); //function 3
    void Register(); //function 4
    void Drive(); //function 5
```

};

```
void main()
{ Race R; }
```

(i) Out of the following, which of the options is correct for calling function 2?

Option 1 - Race T(30); Option 2 - Race U(R);.

(ii) Name the feature of object oriented programming, which is illustrated by function1, function2 and function 3 combined together.

Q5. What is copy constructor? What do you understand by constructor overloading?

Q6. Find the output of following-

<pre>#include<iostream.h> class METRO { int Mno,TripNo,PassengerCount; public: METRO(int Tmno=1) { Mno= Tmno; TripNo =0; PassengerCount=0; } void Trip(int PC=20) { TripNo++; PassengerCount +=PC; } void StatusShow() { cout<<Mno<<":"<<TripNo<<":"<<PassengerCount<<endl; } };</pre>	<pre>void main() { METRO M(5),T; M.Trip(); T.Trip(50); M.StatusShow(); M.Trip(30); T.StatusShow(); M.StatusShow(); }</pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

Unit-I

Object Oriented Programming in C++

Inheritance

Chapter: 04

Inheritance is the process of creating a new class from existing class/classes. The existing class is known as the base/super/parent class and newly created class is known as derived/sub/child class. The derived class will inherit the properties of base class.

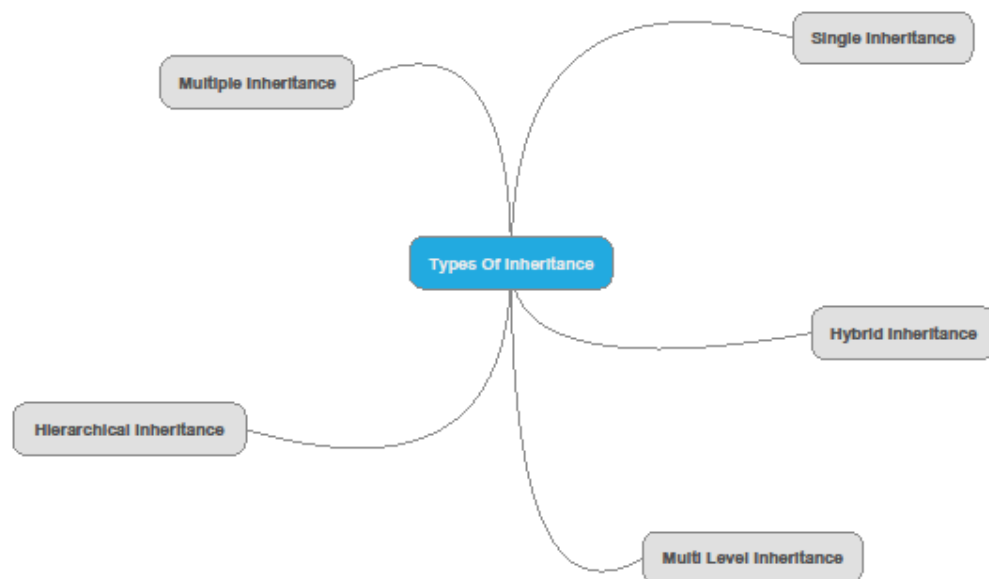
Advantages of Inheritance are given below:

Reusability: It helps the code to be reused in derived class. The base class is defined and once it is compiled, it needs not to be reworked.

Transitivity: If class B inherits properties of another class A, then all subclasses of class B will automatically inherit the properties of A. It is called transitive property.

Types of Inheritance:

1. **Single inheritance:-** When a sub class inherits only from one base class, is known as single inheritance.
2. **Multiple Inheritance:-** When a sub class inherits from multiple base classes, is known as multiple inheritance.
3. **Hierarchical Inheritance:-** When many sub classes inherit from a single class, it is known as hierarchical inheritance.
4. **Multilevel Inheritance:-** When a class inherits from a class that itself inherits from another class it is known as a multilevel inheritance.
5. **Hybrid Inheritance:** It is a combination of 2 or more of above types of inheritance. There is no pattern of deriving from classes.



Syntax for defining a derived class:

```
class<derived class name>:<visibility mode><base class name>
{
```

```
//Data members of derived class
//member functions of derived class
};
```

Visibility modes

The visibility mode in the definition of the derived class specifies whether the features of the base class are privately derived or publicly derived or protected derived.

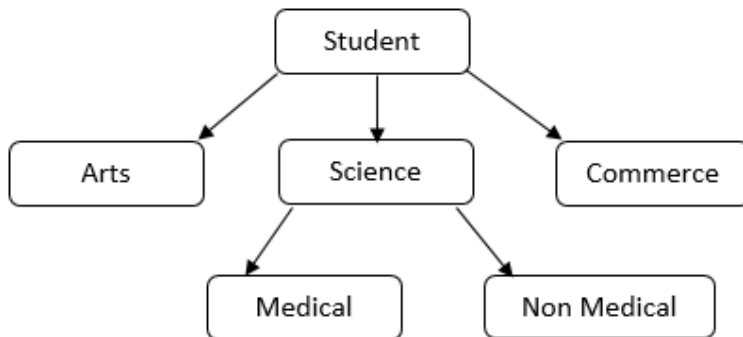
Visibility Mode	Public member of base class becomes	Protected member of base class becomes	
Public	Public in derived class	Protected in derived class	Private member of base class are not directly accessible to derived class
Protected	Protected in derived class	Protected in derived class	
Private	Private in derived class	Private in derived class	

Constructor and Destructor in Derived classes:

When a base class and a derived class both have constructor and destructor , the constructors are executed in order of inheritance and destructors are executed in reverse order. That is, the base constructor is executed before the constructor of the derived class and the destructor of the derived class is executed before the base class destructor.

Short Answer Type Questions(2 Marks)

- Q1.What do you understand by Inheritance?
- Q2.What is Base class and Derived class?
- Q3.What is difference between Multiple Inheritance and Multilevel Inheritance?
- Q4.What is difference between public access specifier and private access specifier?
- Q5.Which type of Inheritance is depicted below?



Q6. Consider the following code and answer the questions below:

```
#include<iostream.h>
class employee
{
    int emp_no;
    char emp_name[25], emp_addr[30], emp_dept[10];
    public:
        void emp_input();
        void emp_print();
};
class manager: public employee
{
    int no_of_emp;
    public:
        void input();
        void print();
};
```

- i. Which type of inheritance is shown in the above code?
- ii. Name the base class and the derived class.
- iii. Name all the private data members of class employee.
- iv. Name all public member functions of class manager.



Unit-I
Object Oriented Programming in C++

Data File Handling In C++

Chapter: 05

File: - The information / data stored under a specific name on a storage device, is called a file.

Stream: - It refers to a sequence of bytes.

Text file: - It is a file that stores information in ASCII characters. In text files, each line of text is terminated with a special character known as EOL (End of Line) character or delimiter character. When this EOL character is read or written, certain internal translations take place.

Binary file:- It is a file that contains information in the same format as it is held in memory. In binary files, no delimiters are used for a line and no translations occur here.

Classes used for different file related operation

ofstream: Object of ofstream class used to write data to the files.

ifstream: Object of ifstream class used to read data from files

fstream: Object of fstream class used to both read and write from/to files.

Opening a file

Opening file using constructor

```
ofstream outFile("sample.txt"); //output only  
ifstream inFile("sample.txt"); //input only
```

Opening File Using open ()

```
StreamObject.open("filename", [mode]);  
ofstream outFile;  
outFile.open("sample.txt");  
ifstream inFile;  
inFile.open("sample.txt");
```

File parameter	mode	Meaning
ios::app		Adds data to the end of file
ios::ate		Cursor goes to end of file on opening and can be placed anywhere in the file.
ios::binary		File opens in binary mode
ios::in		Opens file for reading only
ios::out		Opens file for writing only and delete previous data if file already exists.
ios::nocreate		Open fails if the file does not exist
ios::noreplace		Open fails if the file already exist
ios::trunc		Deletes the contents of the file if it exist

All these flags can be combined using the bitwise operator OR (|). For example, if we want to open the file example.dat in binary mode to add data we could do it by the following call to member function open():

```
fstream file;  
file.open ("example.dat", ios::out | ios::in | ios::binary);
```

Closing File

```
outFile.close();  
inFile.close();
```




Input and output operation

put() and get() function

the function put() writes a single character to the associated stream. Similarly, the function get() reads a single character from the associated stream.

example :

```
file.get(ch);
```

```
file.put(ch);
```

write() and read() function

write() and read() functions write and read blocks of binary data.

example:

```
file.read((char *)&obj, sizeof(obj));
```

```
file.write((char *)&obj, sizeof(obj));
```

Determining End of File.

eof():-returns true (non zero) if end of file is encountered while reading; otherwise return false(zero)

File Pointers And Their Manipulation

All I/O stream objects have, at least, one internal stream pointer: ifstream has a pointer known as the get pointer that points to the element to be read in the next input operation. ofstream has a pointer known as the put pointer that points to the location where the next element has to be written. *fstream, inherits both, the get and the put pointers.* These internal stream pointers that point to the reading or writing locations within a stream can be manipulated using the following member functions:

seekg()	moves get pointer(input) to a specified location
seekp()	moves put pointer (output) to a specified location
tellg()	gives the current position of the get pointer
tellp()	gives the current position of the put pointer

The other prototype for these functions is:

```
seekg(offset, reposition );
```

```
seekp(offset, reposition );
```

The parameter offset represents the number of bytes(any negative or positive integer value for backward or forward movement) the file pointer is to be moved from the location specified by the parameter reposition. The reposition takes one of the following three constants defined in the ios class.

```
ios::beg      start of the file
```

```
ios::cur      current position of the pointer
```

```
ios::end      end of the file
```

<p>Program to count number of words from a text file“input.txt”</p> <pre>#include<fstream.h> void main() { ifstream fin; fin.open("input.txt"); char words[50]; int count=0; while(!fin.eof())</pre>	<p>Program to count number of vowels in a text file “input.txt”</p> <pre>#include<fstream.h> void main() { ifstream fin; fin.open("input.txt"); char ch; int count=0;</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Computer Science

<pre> { fin>>words; count++; } cout<<"Number of words in file is "<<count; fin.close(); } </pre>	<pre> while(!fin.eof()) { fin.get(ch); if(ch=='a' ch=='e' ch=='i' ch=='o' ch=='u') count++; } cout<<"Number of vowels in file are "<<count; fin.close();} </pre>
--------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Binary File input output and modify record

<pre> #include<fstream.h> #include<string.h> #include<stdio.h> class employee { int empno; char ename[20]; public: void getdata() { cout<<"\nEnter The Employee no. "; cin>>empno; cout<<"\nEnter The Name of The Employee "; gets(ename); } void showdata() { cout<<"\nEmployeee no. : "<<empno; cout<<"\nEmployee Name : "; cout<<ename; } int returnempno() { return empno; } }; </pre>	<pre> void write_data() { employee obj; ofstream fout; fout.open("employee.dat",ios::binary/ios::app); obj.getdata(); fout.write((char*)&obj,sizeof(obj)); fout.close(); } void display() { employee obj; ifstream fin; fin.open("employee.dat",ios::binary); while(fin.read((char*)&obj,sizeof(obj))) { obj.showdata(); } fin.close(); } void modifyrecord(int n) { fstream finout; student obj; int found=0; finout.open("employee.dat",ios::in/ios::out/ios::binary); while(finout.read((char*)&obj,sizeof(obj))&& found==0) { if(obj.returnempno()==n) { obj.showdata(); cout<<"\nEnter The New data of employee"; obj.getdata(); int pos=-1*sizeof(obj); finout.seekp(pos,ios::cur); finout.write((char*)&obj,sizeof(obj)); </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre> found=1; } } finout.close(); } </pre>
--	-----------------------------------------------------------------

Short Answer Type Questions(2 Marks based on Text Files)

1. Write a function in a C++ to count the number of lowercase alphabets present in a text file "BOOK.txt".

Ans :-

```

int countalpha()
{
    ifstream Fin("BOOK.txt");
    char ch;
    int count=0;
    while(!Fin.eof())
    {
        Fin.get(ch);
        if (islower(ch))
            count++;
    }
    Fin.close();
    return count;
}
    
```

2. Write a function in C++ to count the number of line started with alphabet 'a' or 'A' in a text file "LINES.TXT".

Ans. :-

```

void counter( )
{
    char Aline[80];
    int Count=0;
    ifstream Fin ("LINES.TXT");
    while(!fin.eof())
    {
        Fin.getline(Aline,80, '\n');
        if (Aline[0]== 'A' || Aline[0]=='a')
            Count++;
    }
    cout<<Count<<endl;
    Fin.close( );
}
    
```

3. Write a function to count number of words whose word length is 8, in a file named "article.txt".

Ans: void wordlen8()

```

{
    char word[20];
    int count=0;
    ifstream fin("article.txt");
    while(!fin.eof())
    {
        fin>>word;
    }
}
    
```

```

        if(strlen(word)==8)
            count++;
    }
    cout<<"Total number of words with word length 8 is=" << count; fin.close();
}

```

4. Given a binary file PHONE.DAT, containing records of the following structure type.

```

class phonlist
{ char Name[20] ;
char Address[30] ;
char AreaCode[5] ;
char PhoneNo[15] ;
public :
void Register( ) ;
void Show( ) ;
int CheckCode(char AC[ ])
{ return strcmp(AreaCode, AC) ;
}
} ;

```

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

Ans: void transfer()

```

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

```

5. Given a binary file STUDENT.DAT, containing records of the following class Student type

```

class Student
{
char S_Admno[10]; //Admission number of student
char S_Name[30]; //Name of student
int Percentage; //Marks Percentage of student
public:
void EnterData()
{
gets(S_Admno); gets(S_Name); cin>>Percentage;
}
void DisplayData()

```

```

{
cout<<setw(12)<<S_Admno;
cout<<setw(32)<<S_Name;
cout<<setw(3)<<Percentage<<endl;
}
int ReturnPercentage(){return Percentage;}
};

```

Write a function in C++, that would read contents of file STUDENT.DAT and display the details of those Students whose Percentage is above 75

Answer :-

```

void Distinction()
{
Student S;
fstream Fin;
Fin.open("STUDENT.DAT", ios::binary|ios::in);
while(Fin.read((char*)&S, sizeof(Student))
if (S.ReturnPercentage())>75)
S.DisplayData( );
Fin.close();
}

```

6. Given a binary file STUINFO.DAT, containing records of the following structure type.

```

class STUDENT
{
int rollno;
char Name[20] ;
char Address[30] ;
char PhoneNo[15] ;
public :
void enter( )
{
cin>>rollno;
cin.getline(name,20);
cin.getline(address,30);
cin.getline(phoneno,15);
}
void display( )
{
cout<<"information of student is";
cout<<rollno<<name<<address<<phoneno;
}
};

```

Write a function stu_write() in C++, that would write information of students in STUINFO.DAT



Very Short Questions (1 Mark based on file pointer)

1. Observe the program segment given below carefully and fill the blanks marked as Line 1 and Line 2 using fstream functions for performing the required task. 1

```
#include <fstream.h>
class Library
{
long Ano; //Ano – Accession Number of the Book
char Title[20]; //Title – Title of the Book
int Qty; //Qty – Number of Books in Library
public:
void Enter(int); //Function to enter the content
void Display(); //Function of display the content
void Buy(int Tqty)
{
Qty+=Tqty;
} //Function to increment in Qty
long GetAno() {return Ano;}
};
void BuyBook (long BANo, int BQty)
//BANo is Accessionno of the book purchased
//BQty is Number of books purchased
{
fstream File;
File. open (“STOCK.DAT”, ios: : binarylios: : inlios: : out);
int Position=-1;
Library L;
while (Position == -1 && File. read ((char*) &L, sizeof (L)))
if (L. GetAno() ==BANo)
{
L. Buy (BQty); //To update the number of Books
Positions=File. tellg()-sizeof (L);
//Line 1: To place the file pointer to the required position.
_____
//Line 2: To write the object L on to the binary file
_____
}
if (Position== -1)
cout<<“No updation done as required Ano not found...”;
File. Close();
}
```

**Ans. : File. seekp (position, ios :: beg); // Line-1
File. write ((char *) & L, sizeof (L)); // Line-2**



Unit-II
Data Structure

Arrays, Stacks, Queues And Linked List

Chapter: 06

In Computer Science, a **data structure** is a particular way of storing and organizing data in a computer so that it can be used efficiently. Different kinds of data structures are suited to different kinds of applications, and some are highly specialized to specific tasks.

Simple Data Structure: These data structures are normally built from primitive data types like integers, floats, characters. For example arrays and structure.

Compound Data Structure: simple data structures can be combined in various ways to form more complex structure called compound structures. Linked Lists, Stack, Queues and Trees are examples of compound data structure.

Searching methods in array

Linear Search: In this method each element of the array is compared with the number to be searched in linear order (from first to last). And where the number is matched the position is displayed.

```
#include<iostream.h>
#include<conio.h>
void main()
{
int lsearch(int[],int,int);
int a[50],item,n,index;
clrscr();
cout<<"\n Enter size of array";
cin>>n;
cout<<"\n Enter array elements";
for(int i=0;i<n;i++)
cin>>a[i];
cout<<"Enter the item to be searched";
cin>>item;
index=lsearch(a,n,item);
if(index== -1)
cout<<"\n Element not found";
else
cout<<"\n Element found at position
"<<index+1;
getch();
}

int lsearch(int a[],int size,int item)
{
for(int i=0;i<size;i++)
{
```

Binary Search Method

Binary search algorithm is applicable for already sorted array only. In this algorithm, to search for the given item from the sorted array (in ascending order), the item is compared with the middle element of the array. If the middle element is equal to the item then index of the middle element is returned, otherwise, if item is less than the middle item then the item is present in first half segment of the array (i.e. between 0 to middle-1), so the next iteration will continue for first half only, if the item is larger than the middle element then the item is present in second half of the array (i.e. between middle+1 to size-1), so the next iteration will continue for second half segment of the array only. The same process continues until either the item is found (search successful) or the segment is reduced to the single element and still the item is not found (search unsuccessful).

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

void main()
{
int bsearch(int[],int,int);
int a[50], item, n, index;
clrscr();
cout<<"\n Enter total elements";
cin>>n;
```

<pre> if(a[i]==item) return i; } return -1; } </pre>	<pre> cout<<"\n Enter array elements in sorted form:"; for(int i=0;i<n;i++) cin>>a[i]; cout<<"Enter the item to be searched"; cin>>item; index=bsearch(a, n, item); if(index== -1) cout<<"\n Element not found"; else cout<<"\n Element found at position "<<index+1; getch(); } int bsearch(int a[], int size, int item) { int beg, last; beg=0; last=size-1; int mid=(last+beg)/2; while(beg<=last) { mid=(beg+last)/2; if(item==a[mid]) { return mid; } else if(item>a[mid]) beg=mid+1; else last=mid-1; } return -1; } </pre>
------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sorting operation in the array

Sorting means to arrange the array elements in Ascending order or in Descending order. There are various methods to do this but for the ease sake Bubble sort method is displayed here.

```

#include<iostream.h>
#include<conio.h>
void bubblesort (int[],int);
void main()
{
int a[50],n;
clrscr();
cout<<"\nHow many elements do you want to create array with? ";
cin>>n;
cout<<"\nEnter array elements\n";

```



```

for(int i=0;i<n;i++)
    cin>>a[i];
bubblesort(a,n);
cout<<"\n\nThe sorted array is as shown below\n";
for(i=0;i<n;i++)
    cout<<a[i]<<"\n";
getch();
}
void bubblesort(int a[],int n) //Function to perform bubble sort
{
    int temp;
    for(int i=0;i<n-1;i++)
    {
        for(int j=0;j<n-i-1;j++)
            if(a[j]>a[j+1])
            {
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
    }
}

```

Some Questions Based on Array (2 marks question)

Q1. Write a function in C++ which accepts an integer array and its size as arguments/parameters and reverses the array

example : if the array is 1,2,3,4,5 then rearrange the array as 5,4,3,2,1

Ans : void reverse(int arr[], int n)

```

{
    int temp;
for(int i=0,j=n-1; i<=j; i++,j--)
    {
        temp= arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
}

```

Q2. Write a function in C++ which accepts an integer array and its size as arguments/parameters and exchange the array in the given manner

example : if the array is 1,2,3,4,5,6,7,8,9,10 then rearrange the array as 2,1,4,3,6,5,8,7,10,9

Ans : void change(int arr[], int n)

```

{
    int temp;
for(int i=0; i<n; i=i+2)
    {
        temp= arr[i];
        arr[i] = arr[i+1];

```

```

arr[i+1] = temp;
}
}

```

Q3 Write a function in C++ to merge the contents of two sorted arrays A & B into third array C. Assuming array A is sorted in ascending order, B is sorted in descending order, the resultant array is required to be in ascending order.

Q 4 Write a function in C++ which accepts an integer array and its size as arguments and assign the elements into a two dimensional array of integers in the following format

If the array is 1,2,3,4,5,6

The resultant 2D array is

```

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

```

if the array is 1,2,3

The resultant 2D array is

```

1 2 3
0 1 2
0 0 1

```

Questions based on Two Dimensional Array(3 marks question)

Q1. Write a function in C++ that will accept a 2-D array and its row and column size as argument and find sum of rows and columns

Ans : void rowcolsum(int A[][],int N, int M)

```

{
    for (int i=0;i<N;i++)
    {
        int SumR=0;
        for (int j=0;j<M;j++)
            SumR+=A[i][j];
        cout<<SumR<<endl;
    }
    for (int i=0;i<N;i++)
    {
        int SumC=0;
        for (int j=0;j<M;j++)
            SumC+=A[j][i];
        cout<<SumC<<endl;
    }
}

```

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

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

```

{
int SumD1=0,SumD2=0;
for (int I=0;I<N;I++)
{
    SumD1+=A[I][I];

```



```
SumD2+=A[N-I-1][I];
}
cout<<"Sum of Diagonal 1:"<<SumD1<<endl;
cout<<"Sum of Diagonal 2:"<<SumD2<<endl;
}
```

Address Calculation in Two Dimensional Array

Two dimensional array can be arranged in two manner

1. Row Major Order
2. Column Major Order

To find the address of a particular row and column the formula in **Row Major Order** is

Address of A[row][column]=B +w*(n(row)+column)

Where

B= Base address of the array

w= Word size

n= total no of columns in the array

To find the address of a particular row and column the formula in **Column Major Order** is

Address of A[row][column]=B +w*(m(Column)+row)

Where

B= Base address of the array

w= Word size

m= total no of rows in the array

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

Ans: Here the array is stored in Row Major Order so

B=4500

W= 4

N= 10

As per the formula

Address of A[row][column]=B +w*(n(row)+column)

=4500+4*(10(12)+8)

=4500+4*(128)

=4500+512

=5012

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

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

Column Major Formula:

Loc(P[I][J]) =Base(P)+W*(N*J+I)

Loc(P[2][20]) =Base(P)+4*(20*20+2)

Base(P) =5000 – 4*(400+2)

=5000 – 1608

=3392



Computer Science

Q3. An array $S[40][30]$ is stored in the memory along the row with each of the element occupying 2 bytes, find out the memory location for the element $S[20][10]$, if an element $S[15][5]$ is stored at the memory location 5500.

Ans. Given, $W=2$, $N=40$, $M=30$, $Loc(S[15][5])=5500$

Row Major Formula:

$$Loc(S[I][J]) = Base(S) + W * (M * I + J)$$

$$Loc(S[15][5]) = Base(S) + 2 * (30 * 15 + 5)$$

$$5500 = Base(S) + 2 * (450 + 5)$$

$$Base(S) = 5500 - 910 = 4590$$

$$\begin{aligned} Loc(S[20][10]) &= 4590 + 2 * (30 * 20 + 10) \\ &= 4590 + 2 * (600 + 10) \\ &= 4590 + 1220 = 5810 \end{aligned}$$

STACKS, QUEUES AND LINKED LIST

Stack

In computer science, a stack is a Last in, First out (LIFO) data structure. It simply means that an element that is inserted at the end will be deleted first. To Manage a stack all the insertion and deletion takes place from one position/end called “top”.

One of the common uses of stack is in function call.

Operations on the Stack

There are two fundamental operations

Push

Pop

Push means to insert an element

Pop means to delete an element

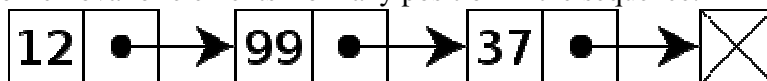
Queue

In computer science, a Queue is a First in, First out (FIFO) data structure. It simply means that an element that is inserted at the beginning will be deleted first. To manage a queue all the insertion and deletion takes place from two different positions called “front” and “rear”.

Every element is inserted from the rear position and deleted from the front position in the queue.

Linked List

A linked list is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of a data and a reference (in other words, a *link*) to the next node in the sequence; more complex variants add additional links. This structure allows for efficient insertion or removal of elements from any position in the sequence.



Here in the figure is an example of a linked list whose nodes contain two fields: an integer value and a link to the next node. The last node is linked to a terminator used to signify the end of the list.

Linked lists are among the simplest and most common data structures. They can be used to implement several other common abstract data types, stacks, queues etc.

The principal benefit of a linked list over an array is that the list elements can easily be inserted or removed without reallocation or reorganization of the entire structure because the data items *need not be stored contiguously* in memory or on disk. Linked lists allow insertion and removal of nodes at any point in the list, and can do so with a constant number of operations if the link previous to the link being added or removed is maintained during list traversal.

Linked list are dynamic structure where memory allocation takes place at run time.

Operation on a linked list

There are three basic operations on a linked list

Insertion

Deletion

Traversal

Inserting a node or element into Linked list :

Inserting an element into linked list contains 3 types .

1. Insertion at beginning of the Linked list
2. Insertion after/before any element of the linked list
3. Insertion at the end of the linked list

Deleting a node from the Linked list.

A node can be deleted in 3 ways similar to Insertion.

1. Deleting a Node from the beginning of the Linked List
2. Deleting a node before/after an element from the Linked list.
3. Deleting a node from the end of the Linked List .

Implementation of stacks using a linked list

The stack which is implemented using linked list is called linked stack or dynamic stack

<pre>#include<iostream.h> #include<conio.h> struct node { int data; node * next; }; class stack { node *top; public: stack() { top=NULL; } void stackpush(); void stackpop(); void displaystack(); };</pre>	<pre>void stack::stackpush() { node *ptr; ptr=new node; cout<<"Enter the element to be pushed"<<endl; cin>>ptr->data; if(top==NULL) ptr->next=NULL; else ptr->next=top; top=ptr; } void stack ::stackpop() { node *ptr; ptr=top; cout<<"The popped element is "<<ptr->data; top=top->next; delete ptr; }</pre>
<pre>void stack :: displaystack() { node *ptr; ptr=top; cout<<"The stack is "<<endl; while(ptr!=NULL) { cout<<ptr->data<<endl; ptr=ptr->next; } }</pre>	<pre>void main() { clrscr(); char ans; stack s1; do { s1.stackpush(); cout<<"wish to continue "<<endl; cin>>ans; }while(ans=='y'); s1.displaystack(); cout<<"Press any key to pop an element"<<endl; getch(); s1.stackpop(); s1.displaystack(); getch(); }</pre>

Implementation of queues using a linked list

The queue which is implemented using linked list is called linked queue or dynamic queue

<pre>#include<iostream.h> #include<conio.h> struct node { int data; node * next; }; class queue { node *front,*rear; public: queue() { rear=front=NULL; } void insqueue(); void delqueue(); void dispqueue(); };</pre>	<pre>void queue::insqueue() { node *ptr; ptr=new node; cout<<"Enter the element to be insert"<<endl; cin>>ptr->data; ptr->next=NULL; if(rear==NULL) front=rear=ptr; else { rear->next=ptr; rear=ptr; } } void queue ::delqueue() { node *ptr; ptr=front; cout<<"The deleted element is "<<ptr->data; if(front==rear) front=rear=NULL; else front=front->next; delete ptr; }</pre>
<pre>void queue :: dispqueue() { node *ptr; ptr=front; cout<<"The queue is "<<endl; while(ptr!=NULL) { cout<<ptr->data<<endl; ptr=ptr->next; } }</pre>	<pre>void main() { clrscr(); char ans; queue q1; do { q1.insqueue(); cout<<"wish to continue "<<endl; cin>>ans; }while(ans=='y'); q1.dispqueue(); cout<<"Press any key to delete an element"<<endl; getch(); q1.delqueue(); q1.dispqueue(); getch(); }</pre>



Computer Science

Some Questions based on Board Examination on Linked stack & Linked Queue

Q1. Write a function in C++ to delete a node containing customer's information, from a dynamically allocated Queue of Customers implemented with the help of the following structure:

```
struct Customer
{int CNo;
    char CName[20];
    Customer *Link;
```

```
};
Ans: struct Customer
```

```
{
    int CNo;
    char CName[20];
    Customer *Link;
} *Front, *Rear, *ptr;
void DELETE()
{
    if(Front == NULL)
        cout<<"\n Queue Underflow\n";
    else
    {
        ptr = Front;
        Front = Front->Link;
        delete ptr;
    }
}
```

Q2. Write a function in C++ to delete a node containing Book's information, from a dynamically allocated Stack of Books implemented with the help of the following structure.

```
struct Book
{int BNo;
    char BName[20];
    Book *Next;
};
```

```
Ans: struct Book
```

```
{
    int BNo;
    char BName[20];
    Book *Next;
} *top, *ptr;
void POP()
{
    if(top == NULL)
        cout<<"\n Stack Underflow\n";
    else
    {
        ptr = top;
        top = top->Next;
        delete ptr;
    }
}
```




Computer Science

Q 3. Evaluate the postfix notation of expression.

4, 10, 5, +, *, 15, 3, /, -

Sno.	Symbol	Stack
0		[
1	4	[4
2	10	[4,10
3	5	[4,10,5
4	+	[4 [4,15
5	*	[[60
6	15	[60,15
7	3	[60,15,3
8	/	[60 [60,5
9	-	[[55
10]	55 Ans

Q. 4. Convert the following infix expression to its equivalent postfix expression, showing the stack contents for each step of conversion.

$$X / Y + U * (V - W)$$

Ans. :- $X / Y + U * (V - W) = ((X / Y) + (U * (V - W)))$

Element	Stack	Postfix
(
(
X		X
/	/	X
Y	/	XY
)		XY/
+	+	XY/
(+	XY/
U	+	XY/U
*	+	XY/U
(+(XY/U
V	+(XY/UV

Computer Science

	+*(-	XY/UV
	+*(-	XY/UVW
	+*	XY/UVW-
X + U*	+	XY/UVW-*
		XY/UVW-*+

(V-W)= XY/UVW-*+

Unit-III

DATABASES MANAGEMENT SYSTEM AND SQL

DBMS & Structured Query Language

Chapter: 07

➤ **Basic Database concepts**

Data : Raw facts and figures which are useful to an organization. We cannot take decisions on the basis of data.

Information: Well processed data is called information. We can take decisions on the basis of information

Field: Set of characters that represents specific data element.

Record: Collection of fields is called a record. A record can have fields of different data types.

File: Collection of similar types of records is called a file.

Table: Collection of rows and columns that contains useful data/information is called a table. A table generally refers to the passive entity which is kept in secondary storage device.

Database: Collection of logically related data along with its description is termed as database.

Relation: Relation (collection of rows and columns) generally refers to an active entity on which we can perform various operations. It is also known as table.

Tuple: A row in a relation is called a tuple. It is also known as record.

Attribute: A column in a relation is called an attribute. It is also termed as field or data item.

Degree: Number of attributes in a relation is called degree of a relation.

Cardinality: Number of tuples in a relation is called cardinality of a relation.

Primary Key: Primary key is a key that can uniquely identifies the records/tuples in a relation. This key can never be duplicated and NULL.

Foreign Key: Non key attribute of a table acting as primary key in some other table is known as Foreign Key in its current table. This key is used to enforce referential integrity in RDBMS.

Candidate Key: Attributes of a table which can serve as a primary key are called candidate keys.

Alternate Key: All the candidate keys other than the primary key of a relation are alternate keys for a relation.

DBA: Data Base Administrator is a person (manager) that is responsible for defining the data base schema, setting security features in database, ensuring proper functioning of the data bases etc.

Select Operation: The select operation selects tuples from a relation which satisfy a given condition. It is denoted by lowercase Greek Letter σ (sigma).

Project Operation: The project operation selects columns from a relation which satisfy a given condition. It is denoted by lowercase Greek Letter π (pi). It can be thought of as picking a sub set of all available columns.

Union Operation: The union (denoted as \cup) of a collection of relations is the set of all distinct tuples in the collection. It is a binary operation that needs two relations.

Set Difference Operation: This is denoted by $-$ (minus) and is a binary operation. It results in a set of tuples that are in one relation but not in another

➤ **Structured Query Language**

SQL is a non procedural language that is used to create, manipulate and process the databases (relations).



1. Data Definition Language (DDL)

DDL contains commands that are used to create, modify or remove the tables, databases, indexes, views, sequences and synonyms etc.

e.g: Create table, create view, create index, alter table, drop table etc.

2. Data Manipulation Language (DML)

DML contains commands that can be used to manipulate the data base objects and to query the databases for information retrieval.

e.g Select, Insert, Delete, Update etc.

3. Transaction Control Language (TCL)

TCL include commands to control the transactions in a data base system. The commonly used commands in TCL are COMMIT, ROLLBACK etc.

➤ **Operators in SQL:** The following are the commonly used operators in SQL

- | | |
|-------------------------|---------------------|
| 1. Arithmetic Operators | +, -, *, / |
| 2. Relational Operators | =, <, >, <=, >=, <> |
| 3. Logical Operators | OR, AND, NOT |

➤ **Data types of SQL:** Just like any other programming language, the facility of defining data of various types is available in SQL also. Following are the most common data types of SQL.

- 1) NUMBER e.g. Number(n,d) Number (5,2)
- 2) CHAR CAHR(SIZE)
- 3) VARCHAR / VARCHAR2 VARCHAR2(SIZE)
- 4) DATE DD-MON-YYYY

➤ **Constraints:** Constraints are the conditions that can be enforced on the attributes of a relation. The constraints come in play whenever we are trying to insert, delete or update a record in a relation.

Not null ensures that we cannot leave a column as null. That is a value has to be supplied for that column.

e.g. name varchar(25) not null

Unique constraint means that the values under that column are always unique.

e.g. Roll_no number(3) unique

Primary key constraint means that a column cannot have duplicate values and not even a null value.

e.g. Roll_no number(3) primary key

The main difference between unique and primary key constraint is that a column specified as unique may have null value but primary key constraint does not allow null values in the column.

Foreign key is used to enforce referential integrity and is declared as a primary key in some other table.

e.g. cust_id varchar(5) references master(cust_id)

it declares cust_id column as a foreign key that refers to **cust_id** field of table **master**.

That means we cannot insert that value in cust_id field whose corresponding value is not present in cust_id field of master table. Moreover we can't delete any row in master table, if a corresponding value of cust_id field is existing in the dependent table.

Check constraint limits the values that can be inserted into a column of a table.



e.g. marks number(3) check(marks>=0)

The above statement declares marks to be of type number and while inserting or updating the value in marks it is ensured that its value is always greater than or equal to zero.

Default constraint is used to specify a default value to a column of a table automatically. This default value will be used when user does not enter any value for that column.

e.g. balance number(5) default = 0

➤ SQL COMMANDS :

1. **Create Table command** is used to create a table . The syntax of this Command is:

```
CREATE TABLE <Table_name>
  ( column_name 1  data_type1 [(size) column_constraints],
    column_name 1  data_type1 [(size) column_constraints],
    :
    :
    [<table_constraint> (column_names)] );
```

2. **The ALTER Table command** is used to change the definition (structure) of existing table.

ALTER TABLE <Table_name>ADD/MODIFY <Column_definition>; For Add or modify column
ALTER TABLE <Table_name> DROP COLUMN <Column_name>; For Deleting a column

3. **The INSERT Command:** The rows (tuples) are added to a table by using INSERT command. The syntax of Insert command is:

```
INSERT INTO <table_name> [(<column_list>)]VALUES (<value_list>);
```

e.g.,

```
INSERT INTO EMP (empno, ename, sex, sal, deptno) VALUES(1001, 'Ravi', 'M', 4500.00, 10);
```

If the order of values matches the actual order of columns in table then it is not required to give the column_list in INSERT command. e.g.

```
INSERT INTO EMP VALUES(1001, 'Ravi', 'M', 4500.00, 10);
```

4. **The Update command** is used to change the value in a table. The syntax of this command is:

```
UPDATE <table_name>
SET column_name1=newvalue1/expression [,column_name2=newvalue2/expression,.....]
WHERE <condition>;
```

e.g., to increase the salary of all the employees of department No 10 by 10% , then command will be:

```
UPDATE emp
SET sal=sal*1.1
WHERE Deptno=10;
```

5. **The DELETE command** removes rows from a table. This removes the entire rows, not individual field values. The syntax of this command is

```
DELETE FROM <table_name>
[WHERE <condition>;]
```

e.g., to delete the tuples from EMP that have salary less than 2000, the following command is used:

```
DELETE FROM emp WHERE sal<2000;
```

To delete all tuples from emp table:

```
DELETE FROM emp;
```

6. **The SELECT command** is used to make queries on database. A query is a command that is given to produce certain specified information from the database table(s). The SELECT command



can be used to retrieve a subset of rows or columns from one or more tables. The syntax of Select Command is:

```
SELECT <Column-list>
FROM <table_name>
[WHERE<condition>]
[GROUP BY <column_list>]
[HAVING<condition>]
[ORDER BY <column_list [ASC|DESC ]>]
```

The **select** clause list the attributes desired in the result of a query.

e.g.,To display the names of all Employees in the *emp* relation:

```
select ename from emp;
```

To force the elimination of duplicates, insert the keyword **distinct** after select.

Find the number of all departments in the *emp* relations, and remove duplicates

```
select distinctdeptno from emp;
```

An asterisk (*) in the select clause denotes “all attributes”

```
SELECT* FROM emp;
```

The **select** clause can contain arithmetic expressions involving the operation, +, −, *, and /, and operating on constants or attributes of tuples.The query:

```
SELECTempno, ename, sal * 12 FROM emp;
```

would display all values same as in the *emp* relation, except that the value of the attribute *sal* is multiplied by 12.

The WHERE clause in SELECT statement specifies the criteria for selection of rows to be returned.

- **Conditions based on a range (BETWEEN Operator):**The Between operator defines a range of values that the column values must fall in to make condition true . The range includes both lower value and upper value.

e.g., Find the empno of those employees whose salary between 90,000 and 100,000 (that is, 90,000 and 100,000)

```
SELECT empno FROM emp WHERE salBETWEEN 90000 AND 100000;
```

- **Conditions based on a list (IN operator):** To specify a list of values , IN operator is used. IN operator selects values that match any value in a given list of values.

For example , to display a list of members from ‘DELHI’, ‘MUMBAI’, ‘CHENNAI’ or ‘BANGALORE’ cities :

```
SELECT * FROM members WHERE city IN (‘DELHI’, ‘MUMBAI’, ‘CHENNAI’, ‘BANGALORE’);
```

The **NOT IN** operator finds rows that do not match in the list. So if you write **SELECT * FROM** members WHERE city **NOT IN** (‘DELHI’, ‘MUMBAI’, ‘CHENNAI’, ‘BANGALORE’);

It will list members not from the cities mentioned in the list.

- **Conditions based on Pattern:** SQL also includes a string-matching operator, LIKE, for comparison on character string using patterns. Patterns are described using two special wildcard characters:

- ➔ Percent (%) – ‘%’ matches any substring(one,more than one or no character).
- ➔ Underscore (_) – ‘_’ character matches exactly one character.

- ☐ Patterns are case-sensive.
- ☐ **Like** keyword is used to select row containing columns that match a wildcard pattern.
- ☐ The keyword **not like** is used to select the row that do not match the specified patterns of characters.

- **Searching for NULL:**The NULL value in a column is searched for in a table using *IS NULL* in the WHERE clause(Relational Operators like =,<> etc can not be used with NULL).



For example, to list details of all employees whose departments contain NULL (i.e., novalue), you use the command:

```
SELECT empno, ename FROM emp Where Deptno IS NULL;
```

- **ORDER BY Clause:** Whenever a select query is executed the resulting rows are displayed in the order in which they exist in the table. You can sort the result of a query in a specific order using ORDER BY clause. The ORDER BY clause allows sorting of query result by one or more columns. The sorting can be done either in ascending or descending order.

Note:- If order is not specified that by default the sorting will be performed in ascending order.

- **GROUP BY Clause:** The GROUP BY clause groups the rows in the result by columns that have the same values. Grouping is done on column name. It can also be performed using aggregate functions in which case the aggregate function produces a single value for each group.
- **Aggregate Functions:** These functions operate on the multiset of values of a column of a relation, and return a value

avg: average value
min: minimum value
max: maximum value
sum: sum of values
count: number of values

These functions are called aggregate functions because they operate on aggregates of tuples. The result of an aggregate function is a single value.

- **HAVING Clause:** The HAVING clause places conditions on groups in contrast to WHERE clause that places conditions on individual rows. While WHERE condition cannot include aggregate functions, HAVING conditions can do so. e.g.,

```
Select deptno, avg(sal), sum(sal) from emp group by deptno having deptno=10;
```

```
Select job, count(*) from emp group by job having count(*)<3;
```

7. **The DROP Command :** The DROP TABLE command is used to drop (delete) a table from database. But there is a condition for dropping a table ; it must be an empty table i.e. a table with rows in it cannot be dropped. The syntax of this command is :

```
DROP TABLE <Table_name>;
```

e.g.,

```
DROP TABLE EMP;
```

8. **Query Based on Two table (Join):**

```
SELECT <Column-list>  
FROM <table_name1>, <table_name2>  
WHERE <Join_condition>[AND condition];
```

9. **Qualified Names :**

```
<tablename>.<fieldname>
```

This type of field names are called qualified field names and are used to identify a field if the two joining tables have fields with the same name.

6 – Marks Questions

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

Table: GAMES

GCode	GameName	Number	PrizeMoney	ScheduleDate
101	Carom Board	2	5000	23-Jan-2004
102	Badminton	2	12000	12-Dec-2003
103	Table Tennis	4	8000	14-Feb-2004



Computer Science

105	Chess	2	9000	01-Jan-2004
108	Lawn Tennis	4	25000	19-Mar-2004

Table: PLAYER

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

- (i) To display the name of all Games with their Gcodes.
- (ii) To display details of those games which are having PrizeMoney more than 7000.
- (iii) To display the content of the GAMES table in ascending order of ScheduleDate.
- (iv) To display sum of PrizeMoney for each of the Number of participation groupings (as shown in column Number 2 or 4).
- (v) SELECT COUNT(DISTINCT Number) FROM GAMES;
- (vi) SELECT MAX(ScheduleDate),MIN(ScheduleDate) FROM GAMES;
- (vii) SELECT SUM(PrizeMoney) FROM GAMES;
- (viii) SELECT DISTINCT Gcode FROM PLAYER;

- Ans :
- (i) SELECT GameName,Gcode FROM GAMES;
 - (ii) SELECT * FROM GAMES WHERE PrizeMoney>7000;
 - (iii) SELECT * FROM GAMES ORDER BY ScheduleDate;
 - (iv) SELECT SUM(PrizeMoney),Number FROM GAMES GROUP BY Number;
 - (v) 2
 - (vi) 19-Mar-2004 12-Dec-2003
 - (vii) 59000
 - (viii) 101
103
108

Q2. Consider the following tables FACULTY and COURSES. Write SQL commands for the statements (i) to (v) and give outputs for SQL queries (vi) to (vii).

FACULTY

F_ID	Fname	Lname	Hire_date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

COURSES

C_ID	F_ID	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Basic	6000

- i) To display details of those Faculties whose salary is greater than 12000.
- ii) To display the details of courses whose fees is in the range of 15000 to 50000 (both



Computer Science

values included).

iii) To increase the fees of all courses by 500 of “System Design” Course.

iv) To display details of those courses which are taught by ‘Sulekha’ in descending order of courses.

v) Select COUNT(DISTINCT F_ID) from COURSES;

vi) Select Fname,Cname from FACULTY,COURSES where COURSES.F_ID =FACULTY.F_ID;

Ans.: (i) Select * from faculty where salary > 12000;

(ii) Select * from Courses.where fees between 15000 and 50000;

(iii) Update courses set fees = fees + 500 where Cname = “System Design”;

(iv) Select * from faculty fac,courses cour where fac.f_id = cour.f_id and fac.fname = 'Sulekha' order by cname desc;

(v) 4

(vi)

Fname	Cname
Amit	Grid Computing
Rakshit	Computer Security
Rashmi	Visual Basic
Sulekha	Human Biology

2- Marks Questions

Define the following terms with example:

(i) DDL

(ii) DML

(iii) Primary Key

(iv) Candidate Key

(v) Alternate Key

(vi) Foreign Key

(vii) Cardinality of relation

(viii) Degree of relation

(ix) Relation

(x) Attribute

(xi) Tuple

Unit-IV Boolean Algebra

Boolean Algebra

Chapter: 08

Truth table:

Truth table is a table, which represents all the possible values of logical variables/statements along with all the possible results of given combinations of values.

Logical Operators:

Logical operators are derived from the Boolean algebra, which is the mathematical representation of the concepts without going into the meaning of the concepts.

1. NOT Operator—Operates on single variable. It gives the complement value of variable.

2. OR Operator -It is a binary operator and denotes logical Addition operation and is represented by "+" symbol

3. AND Operator – AND Operator performs logical multiplications and symbol is (.) dot.

Truth table:

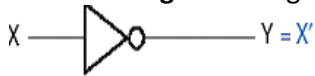
Basic Logic Gates

A gate is simply an electronic circuit, which operates on one or more signals to produce an output signal. Gates are digital circuits because the input and output signals are either low (0) or high (1). Gates also called logic circuits.

There are three types of logic gates:

1. Inverter (NOT gate)
2. OR gate
3. AND gate

1. NOT gate : This gate takes one input and gives a single output. The symbol of this logic gate is



This circuit is used to obtain the compliment of a value.

If $X = 0$, then $X' = 1$.

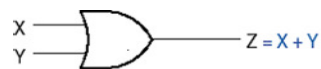
The truth table for NOT gate is :

X	X'
0	1
1	0

2. OR gate : The OR gate has two or more input signals but only one output signal if any of the input signal is 1(high) the output signal is 1(high).

Truth Table and circuit diagram for Two Input OR gate is :

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



AND gate The AND gate have two or more than two input signals and produce an output signal. When all the inputs are 1(High) then the output is 1 otherwise output is 0 only.

Truth Table and circuit diagram for Two Input AND gate is :

X	Y	F=X.Y
0	0	0
0	1	0
1	0	0
1	1	1





Principle of Duality

This principle states that we can derive a Boolean relation from another Boolean relation by performing simple steps. The steps are:-

1. Change each AND(.) with an OR(+) sign
2. Change each OR(+) with an AND(.) sign
3. Replace each 0 with 1 and each 1 with 0

e.g $0+0=0$ then dual is $1.1=1$, $1+0=1$ then dual is $0.1=0$

Basic theorem of Boolean algebra

Basic postulates of Boolean algebra are used to define basic theorems of Boolean algebra that provides all the tools necessary for manipulating Boolean expression.

1. Properties of 0 and 1

$$0+X=X \quad 1+X=1 \quad 0.X=0 \quad 1.X=X$$

2. Idempotence Law

$$X+X=X \quad X.X=X$$

3. Involution Law $(X')' = X$

4. Complementarity Law $X' + X=1 \quad X.X=0$

5. Commutative Law $X+Y=Y+X \quad X.Y=Y.X$

6. Associative Law $X+(Y+Z)=(X+Y)+Z \quad X(YZ)=(XY)Z$

7. Distributive Law $X(Y+Z)=XY_XZ \quad X+YZ=(X+Y)(X+Z)$

8. Absorption Law $X+XY= X \quad X(X+Y)=X$

Demorgan's First Theorem:

This rule states that the compliment of OR of two operands is same as the AND of the compliments of those operands.

Mathematically it can be written as:- $(A+B)'=A'.B'$

Demorgan's Second Theorem:

This rule states that the compliment of AND of two operands is same as the OR of the compliments of those operands.

Mathematically it can be written as:- $(A.B)'=A'+B'$

Algebraic proof of De Morgan's Theorem (First)

$(a+b) + (a'b') = 1$	$(a+b)(a'b') = 0.$
First Part $(a+b)+(a'b')$ $=(a+b+a')(a+b+b')$ (Distribution Law) $= (1+b)(a+1)$ (Complement law) $=1$	Second Part :- $(a+b)(a'b')$ $=(a'b')(a+b)$ (Commutative law) $=a'b'a+a'b'b$ (Distribution Law) $=0*b'+a'*0$ ($x*0=0$) $=0+0$ $=0$

Note: DeMorgan's Second theorem is just the complement of the First Theorem

Minterms and Maxterms

- *Minterm is the product of all the literals with or without bar within a logical system viz if we have two literals A and B then the possible minterms can be AB,A'B,AB',A'B'.*
- *Maxterm is the sum of all the literals with or without bar within a logical system viz if we have two literals A and B then the possible maxterms can be A+B,A'+B,A+B',A'+B'.*

n Variables can be combined to form 2^n minterms or maxterms.

Minterms and Maxterms for Three Binary Variables						
			Minterms		Maxterms	
x	y	Z	Term	Shorthand Notation	Term	Shorthand Notation

Computer Science

0	0	0	$x'y'z'$	m_0	$x+y+z$	M_0
0	0	1	$x'y'z$	m_1	$x+y+z'$	M_1
0	1	0	$x'yz'$	m_2	$x+y'+z$	M_2
0	1	1	$x'yz$	m_3	$x+y'+z'$	M_3
1	0	0	$xy'z'$	m_4	$x'+y+z$	M_4
1	0	1	$xy'z$	m_5	$x'+y+z'$	M_5
1	1	0	xyz'	m_6	$x'+y'+z$	M_6
1	1	1	xyz	m_7	$x'+y'+z'$	M_7

- Sum of Product (SOP) Function: A Boolean function may be represented algebraically from a given truth table by forming a minterm for each combination of the variables that produces a 1 in the function and then taking the OR of all those terms.
- Product of Sum (POS) Function: A Boolean function may be represented algebraically from a given truth table by forming a maxterm for each combination of the variables that produces a 0 in the function and then taking the AND of all those terms.

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

For result F(SOP form is) = $x'y'z + xy'z' + xyz$

For result F(POS form is) = $(x+y+z) \cdot (x+y'+z) \cdot (x+y'+z') \cdot (x'+y+z)$

Example: Express the Boolean function $F(A,B,C) = AB + C$ as a sum of minterms.

Step 1 – Each term must contain all variables

$$AB = AB(C + C') = ABC + ABC'$$

$$C = C(A + A') = AC + A'C$$

$$= AC(B + B') + A'C(B + B')$$

$$= ABC + AB'C + A'BC + A'B'C$$

Step 2 – OR all new terms, eliminating duplicates

$$F(A,B,C) = A'B'C + A'BC + AB'C + ABC' + ABC$$

$$= m_1 + m_3 + m_5 + m_6 + m_7$$

$$= \Sigma(1, 3, 5, 6, 7)$$

Example: Express the Boolean function $F(x,y,z) = x'y + xz$ as a product of maxterms.

Step 1 – Convert the function into OR terms using the distributive law

$$F(x,y,z) = (x'y + x)(x'y + z)$$

$$= (x + x')(y + x)(x' + z)(y + z)$$

$$= (y + x)(x' + z)(y + z)$$

Step 2 – Each term must contain all variables

$$y + x = y + x + zz' = (x + y + z)(x + y + z')$$

$$x' + z = x' + z + yy' = (x' + y + z)(x' + y' + z)$$

$$y + z = y + z + xx' = (x + y + z)(x' + y + z)$$

step 3 – AND all new terms, eliminating duplicates

$$\begin{aligned}
 F(x,y,z) &= (x + y + z)(x + y + z')(x' + y + z)(x' + y' + z) \\
 &= (M_0 M_1 M_4 M_6) \\
 &= \Pi(0, 1, 4, 6)
 \end{aligned}$$

Conversion between Canonical Forms

The complement of a function expressed as the sum of minterms equals the sum of minterms missing from the original function. This is because the original function is expressed by those minterms that make the function equal to 1, whereas its complement is a 1 for those minterms that the function is 0.

Example : $F(A,B,C) = \Sigma(0, 2, 4, 6, 7)$

$$F'(A,B,C) = \Sigma(1, 3, 5) = m_1 + m_3 + m_5$$

Take the complement of F' by DeMorgan's theorem to obtain F in a different form:

$$F(A,B,C) = (m_1 + m_3 + m_5)' = (m_1' \cdot m_3' \cdot m_5') = M_1 M_3 M_5 = \Pi(1, 3, 5)$$

- To convert from one canonical form to the other, interchange the symbols Σ (Sigma) and Π (Pie), and list those numbers missing from the original form.

Minimization of Boolean expressions:-

After obtaining SOP and POS expressions, the next step is to simplify the Boolean expression. There are two methods of simplification of Boolean expressions.

1. Algebraic Method
2. Karnaugh Map :

1. Algebraic method: This method makes use of Boolean postulates, rules and theorems to simplify the expression.

Example. Simplify $AB'CD + A'BCD' + ABCD + ABCD'$

Solution-- $AB'CD + AB'CD' + ABCD + ABCD'$

$$\begin{aligned}
 &= AB'C(D+D') + ABC(D+D') \\
 &= AB'C \cdot 1 + ABC \cdot 1 && (D+D'=1) \\
 &= AC(B'+B) \\
 &= AC \cdot 1 = AC
 \end{aligned}$$

2. Using Karnaugh Map :

A Karnaugh map is graphical display of the fundamental products in a truth table.

For example:

- Put a 1 in the box for any minterm that appears in the SOP expansion.
- Basic idea is to cover the **largest adjacent** blocks you can whose side length is some power of 2.
- Blocks can wrap around (Map rolling) the edges.
- Redundant groups should be avoided.

Sum Of Products Reduction using K- Map

For reducing the expression first mark Octet, Quad, Pair then single.

- Pair: Two adjacent 1's makes a pair. *Pair removes one variable.*
- Quad: Four adjacent 1's makes a quad. *Quad removes two variables.*
- Octet: Eight adjacent 1's makes an Octet. *Octet removes three variables.*

Reduction of expression: When moving vertically or horizontally in pair or a quad or an octet it can be observed that only one variable gets changed that can be eliminated directly in the expression.

For Example

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

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

	CD				
AB		C'D'	C'D	CD	CD'
A'B'		1	1	1	1
A'B		1	1	1	1
AB		1	1	1	1
AB'		1	1	1	1

There are 1 octet, 2 quads after eliminating the redundant groups.

Octet ($m_0, m_2, m_4, m_6, m_8, m_{10}, m_{12}, m_{14}$) reduces to D'

Quad (m_2, m_3, m_6, m_7) reduces to $A'C$

Quad (m_8, m_9, m_{12}, m_{13}) reduces to AC'

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

Product of Sums Reduction using K-Map

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

$$F(A, B, C, D) = \pi (0, 3, 5, 6, 7, 8, 11, 15)$$

	C+D				
A+B		C+D	C+D'	C'+D'	C'+D
A+B		0	0	0	0
A+B'		0	0	0	0
A'+B'		0	0	0	0
A'+B		0	0	0	0

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

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

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

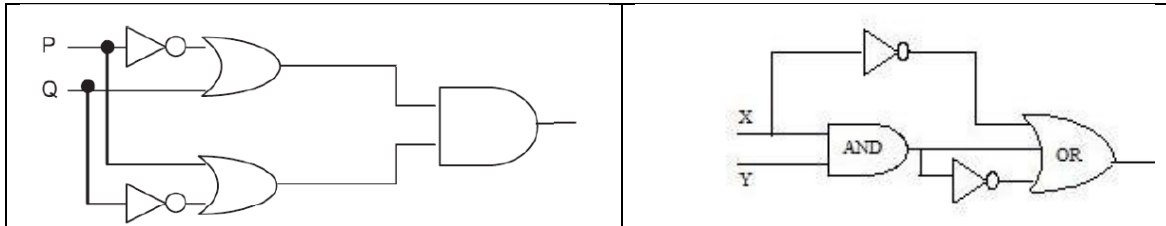
Pair (M_6, M_7) reduces to $A + B' + C'$

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

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

2 Marks Questions

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



2. Draw a Logical Circuit Diagram for the following Boolean expression:

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

3. Prove $x'.y'+y.z = x'yz+x'yz'+xyz+x'yz$ algebraically.

3 Marks Questions

1. If $F(a,b,c,d)=\sum(0,2,4,5,7,8,10,12,13,15)$, obtain the simplified form using K-Map.

2. Obtain a simplified form for a boolean expression

$$F(U,V,W,Z)=\pi(0,1,3,5,6,7,10,14,15)$$

Unit-V Network and Communication Technology

Network and Communication Technology

Chapter: 09

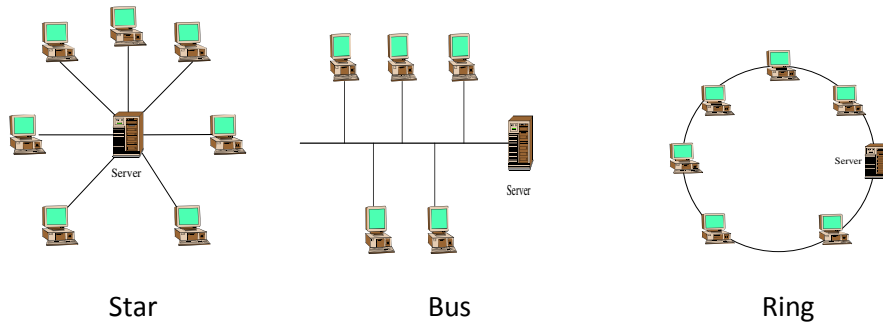
A Network is a collection of interconnected autonomous computers. A Computer which is a part of a network is known as Node or Workstation. The benefits of a network are Resource Sharing, Increased Reliability, Cost Reduction and Effective Communication.

Types of Network: Following types of network exist

- 1 LAN (Local Area Network)
2. MAN (Metropolitan Area Network)
3. WAN (Wide Area Network)
4. PAN (Personal Area Network)

Topology:

It is termed as the physical arrangement of computers in a network viz Bus, Ring, Star, Mesh



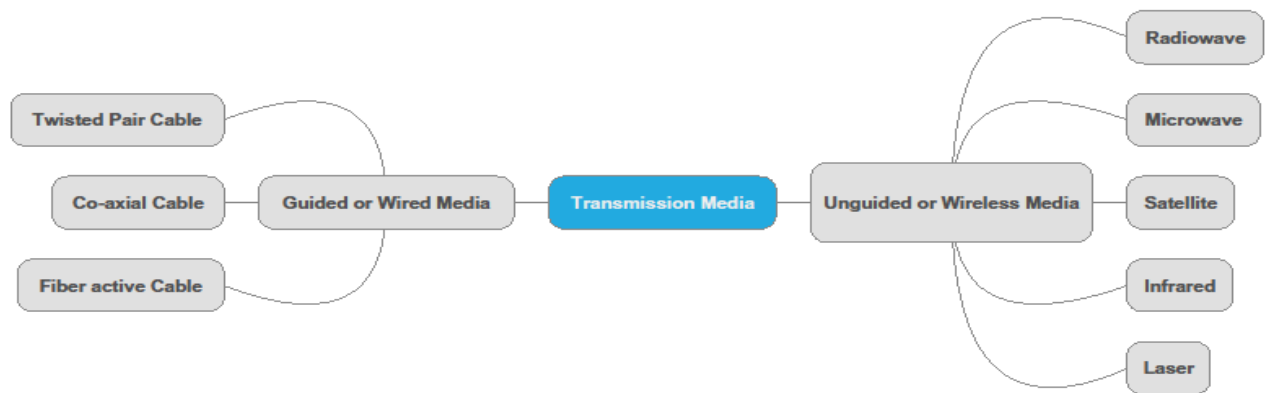
Protocol

Set of rules to govern communication between two computers in a network viz TCP/IP, PPP etc

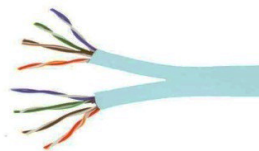
DATA COMMUNICATION TERMINOLOGIES

Data channel	<ul style="list-style-type: none"> The information / data carry from one end to another in the network by channel.
Baud & bits per second (bps)	<ul style="list-style-type: none"> It's used to measurement for the information carry of a communication channel. Measurement Units are Kbps (kilobits Per Second), KBPS (Kilo Byte Per Second), and Mbps (Mega bits Per Second)
Bandwidth	<ul style="list-style-type: none"> It is amount of information transmitted or receives per unit time.

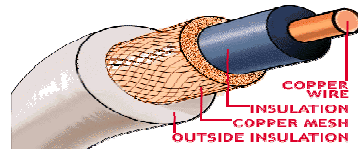
Transmission Media: The **means or channel** through which we send our data from one place to another.



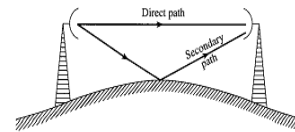
Twisted Pair Cable



Co Axial Cable



Microwaves



Networking Devices

Modem

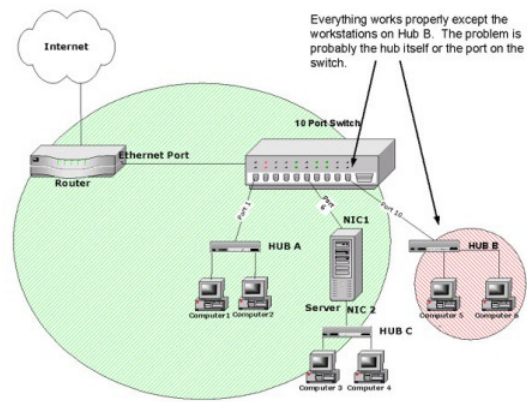
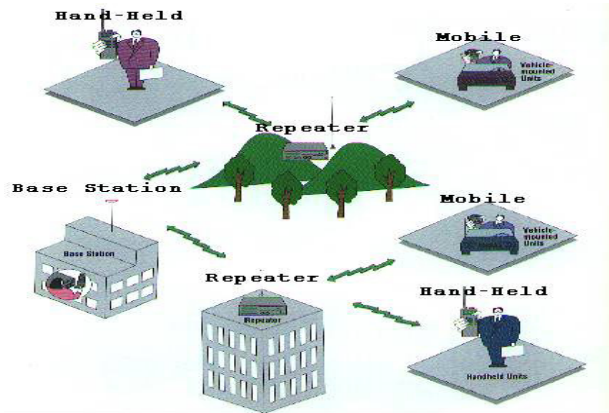
A modem is a computer peripheral that allows you to connect and communicate with other computers via telephone lines. Modem means Modulation/ Demodulation. Modem can be Internal and External.

RJ- 45 Connector

RJ-45 is short for Registered Jack-45. It is an eight wire connector which is commonly used to connect computers on the local area networks i.e., LAN.

Network Interface Cards (Ethernet Card)

A network card, network adapter or NIC (network interface card) or NIU is a piece of computer hardware designed to allow computers to communicate over a **computer network**.



Repeaters:

A repeater is an electronic device that receives a signal and retransmits it at a higher level or higher power, or onto the other side of an obstruction, so that the signal can cover longer distances without degradation.

Hub

A hub contains multiple ports. When a packet arrives at one port, it is copied to all the ports of the hub (i.e. in broadcast manner).



Switch: Switch is a device that performs switching. It forwards chunk of data communication between ports (connected cables) based on the Mac-Addresses in the packets. This is distinct from a hub in that it only forwards the data to the ports involved in the communications rather than all ports connected. The network can be connected directly to a switch, or another switch that is in turn connected to a switch.

iDifference between Bridges, Routers and Gateway

All these network devices are used to connect two or more network or network segments with the following conditions

Bridge	Router	Gateway
Similar Topology and similar protocol	Similar Topology and Different protocol	Different Topology and Different protocol

Common Networking Terms and their expansion

Term	Expansion
TCP/IP	Transmission Control Protocol/Internet Protocol
FTP	File Transfer Protocol
SLIP	Serial Line Internet Protocol
POP	Post Office Protocol
PPP	Point to Point Protocol
SMTP	Simple Mail Transfer Protocol
VoIP	Voice over Internet Protocol
HTTP	Hyper Text Transfer Protocol
URL	Uniform Resource Locator

Computer Science

HTML	Hyper Text Markup Language
DHTML	Dynamic Hyper Text Markup Language
WLL	Wireless in Local loop
GSM	Global System for Mobile Communication
CDMA	Code Division Multiple Access
TDMA	Time Division Multiple Access
SIM	Subscriber's Identity Module
SMS	Short Messaging Service
MMS	Multimedia Messaging Service
EDGE	Enhanced Data rates for Global Evolution
IMAP	Internet Message Access Protocol
Wi-Fi	Wireless Fidelity
IPR	Intellectual Property Rights
WWW	World Wide Web
XML	eXtensible Markup Language
PHP	Hyper Text Preprocessor
ASP	Active Server Pages
JSP	Java Server Pages
FLOSS	Free Libre and Open Source Software
GNU	GNU's Not Unix
OSI	Open Source Initiative
FSF	Free Software Foundation
W3C	World Wide Web Consortium

Some Definitions:-

Hacker:- A Hacker is a programmer who intrudes in a secure network just for gaining knowledge or playful pranks.

Cracker:- A cracker is a malicious programmer who breaks into a secure system with a malafide intention.

Cookies:- Cookies are messages sent by web server to keep track of user's activity.

Web Server:- A web server is a computer system that processes requests of the client via HTTP.

Web Browser:- A web browser (commonly referred to as a browser) is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. e.g. Internet Explorer, Mozilla Firefox, Google Chrome etc.

Firewall:- A firewall is a network security system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both.

Cyber Crime:- Criminal activities carried out by means of computer or Internet. All such crime are dealt with Indian IT Act 2000.

CyberLaw:- Cyberlaw is a generic term used to refer rules for preventing crime on Internet.

Cloud computing:- Cloud Computing is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. The cloud aims to cut costs, and helps the users focus on their core business instead of being impeded by IT obstacles.

Spam:- Irrelevant or unsolicited messages sent over the Internet, typically to large numbers of users, for the purposes of advertising, phishing, spreading malware, etc.



Virus:- A computer virus is a program that, when executed, replicates itself or infecting other programs by modifying them. Infecting computer programs can include data files, or the boot sector of the hard drive.

Antivirus:- Antivirus or anti-virus software, is computer software used to prevent, detect and remove malicious software. Some common example of Antivirus Software are Norton, Quickheal, Kaspersky etc.

OPEN SOURCE TERMINOLOGIES

TERMINOLOGY & DEFINITIONS:

- **Free Software:** They are freely accessible and can be freely used, changed, improved, copied and distributed by all and payments are not needed for free Software.
- **Open Source Software:** Software whose source code is available to the user and it can be modified and redistributed without any limitation .OSS may come free of cost but nominal charges have to be paid for support of Software and development of Software.
- **Proprietary Software:** Proprietary Software is neither open nor freely available, normally the source code of the Proprietary Software is not available but further distribution and modification is possible by special permission by the developer.
- **Freeware:** Freeware are the software freely available , which permit redistribution but not modification (their source code is not available). Freeware is distributed in *Binary Form* (ready to run) without any licensing fees.
- **Shareware:** Software for which license fee is payable after some time limit, its source code is not available and modification to the software are not allowed.

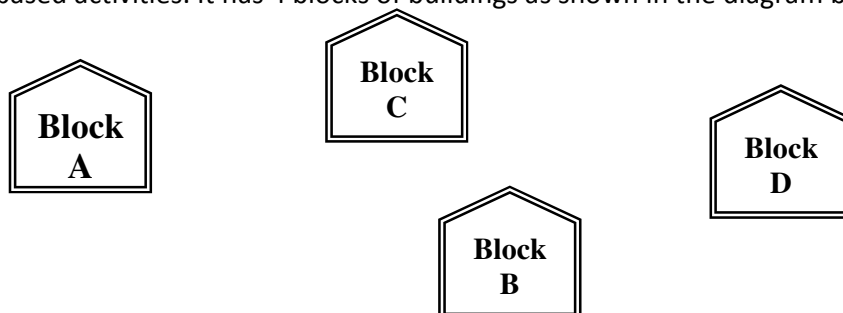
OPEN SOURCE / FREE SOFTWARE

- **Linux :** Linux is a famous computer operating system . Popular Linux server set of program – LAMP(Linux, Apache, MySQL, PHP)
- **Mozilla :** Mozilla is a free internet software that includes
 - a web browser
 - an email client
 - an HTML editor
 - IRC client
- **Apache server:** Apache web server is an open source web server available for many platforms such as BSD, Linux, and Microsoft Windows etc.
 - Apache Web server is maintained by open community of developers of Apache software foundation.
- **MYSQL :** MYSQL is one of the most popular open source database system.
- **OpenOffice :** OpenOffice is an office applications suite. It is intended to compatible and directly compete with Microsoft office.
It includes:
 - Writer (word processor)
 - Calc(spreadsheet)
 - Draw(graphics program)
- **Python: Python** is an interactive programming language originally as scripting language for Amoeba OS capable of making system calls.

Computer Science

Long Answer Questions(4 marks question: 1 mark each)

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



Centre to centre distances between various blocks

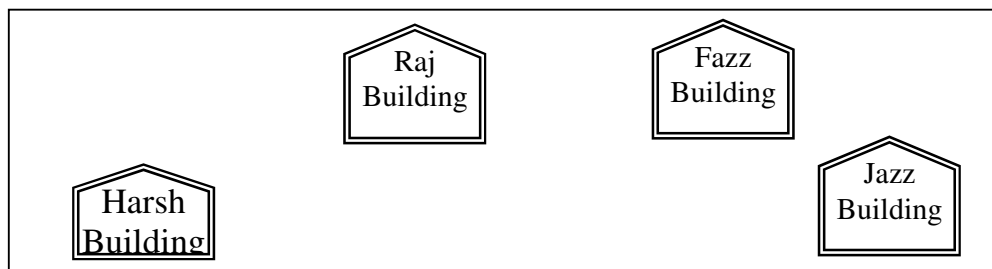
Block A to Block B	50 m
Block B to Block C	150 m
Block C to Block D	25 m
Block A to Block D	170 m
Block B to Block D	125 m
Block A to Block C	90 m

Number of Computers

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

- (a) Suggest a cable layout of connections between the blocks.
- (b) Suggest the most suitable place (i.e. block) to house the server of this organisation with a suitable reason.
- (c) Suggest the placement of the following devices with justification
 - (i) Repeater
 - (ii) Hub/Switch
- (d) The organization is planning to link its front office situated in the city in a hilly region where cable connection is not feasible, suggest an economic way to connect it with reasonably high speed?

Q2. Ravya Industries has set up its new center 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 buildings

Harsh Building to Raj Building	50 m
Raj Building to Fazz Building	60 m
Fazz Building to Jazz Building	25 m
Jazz Building to Harsh Building	170 m
Harsh Building to Fazz Building	125 m
Raj Building to Jazz Building	90 m

Number of Computers

Harsh Building	15
Raj Building	150
Fazz Building	15
Jazz Building	25

- (a) Suggest a cable layout of connections between the buildings.
- (b) Suggest the most suitable place (i.e. building) to house the server of this organisation with a suitable reason.
- (c) Suggest the placement of the following devices with justification:
 - (i) Internet Connecting Device/Modem
 - (ii) Switch
- (d) The organization is planning to link its sale counter situated in various parts of the same city, which type of network out of LAN, MAN or WAN will be formed? Justify your answer.

Very short Answer Questions (1 mark question)

1. Differentiate between hackers and crackers.
2. What is cookies?
3. Define Firewall.
4. Name some open source software.
5. Differentiate between Freeware and free software.
6. How proprietary software is different from shareware?
7. What is cyber law?
8. Define spam.
9. What is Cloud computing?
10. Differentiate between hub and switch.
11. What is web 2.0?
12. Differentiate between worms and virus.
13. Name two client side scripting.



Computer Science

AISSCE-2014 COMPUTER SCIENCE

Time allowed : 3 hours

Maximum Marks : 70

1.(a) Explain in brief the purpose of function prototype with the help of a suitable example. 2

(b) Name the header files that shall be needed for successful compilation of the following C++ code : 1

void main()

```
{ char str[20],str1[20];  
  gets(str); strcpy(str1,str); strrev(str); puts(str); puts(str1); }
```

(c) Deepa has just started working as a programmer in STAR SOFTWARE company. In the company she has got her first assignment to be done using a C++ function to find the smallest number out of a given set of numbers stored in a one-dimensional array. But she has committed some logical mistakes while writing the code and is not getting the desired result. Rewrite the correct code underlining the corrections done. Do not add any additional statements in the corrected code. 2

```
int find(int a[],int n)  
{ int s=a[0];  
  for(int x=1;x<n;x++)  
    if(a[x]>s)  
      a[x]=s;  
  return(s);  
}
```

(d) Find output of the following program segment : 2

```
#include<iostream.h>  
#include<ctype.h>  
void Mycode(char Msg[],char CH)  
{  
  for(int cnt=0;Msg[cnt]!='\0';cnt++)  
    { if(Msg[cnt]>='B'&& Msg[cnt]<='G') Msg[cnt]=tolower(Msg[cnt]);  
      else  
        if(Msg[cnt]=='N' || Msg[cnt]=='n' || Msg[cnt]=='')  
          Msg[cnt]=CH; else  
            if(cnt%2==0)  
              Msg[cnt]=toupper(Msg[cnt]);  
            else  
              Msg[cnt]=Msg[cnt-1]; }  
}
```

```
void main()  
{ char MyText[]="Input Raw";  
  Mycode(MyText,'@');  
  cout<<"NEW TEXT:"<<MyText<<endl; }
```

(e) Find the output of the following program : 3

```
#include<iostream.h>  
void in(int x,int y, int &z)  
{ x+=y;  
  y- -;
```

```

z*=(x-y);
void out(int z,int y, int &x)
{ x*=y;      y++;
z/=(x+y);    }
void main()
{   int a=20, b=30, c=10;
    out(a,c,b);
    cout<<a<<"#"<<b<<"#"<<c<<"#"<<endl;
    in(b,c,a);
    cout<<a<<"@"<<b<<"@"<<c<<"@"<<endl;
    out(a,b,c);
    cout<<a<<"$"<<b<<"$"<<c<<"$"<<endl;
}

```

(f) Write a user defined function DIVT() which takes an integer as parameter and returns whether it is divisible by 13 or not. The function should return 1 if it is divisible by 13, otherwise it should return 0. 2

2.(a) Explain data hiding with an example. 2

(b) Define a class **CONTEST** in C++ with the following description : 4

Private Data Members

Eventno	integer
Description	char(30)
Score	integer
qualified	char

Public Member functions

- A constructor to assign initial values Eventno as 11, Description as "School level", Score as 100, qualified as 'N'.
- Input() – To take the input for Eventno, description and score.
- Award (int cutoffscore) – To assign qualified as 'Y', if score is more than the cutoffscore that is passed as argument to the function, else assign qualified as 'N'.
- Displaydata() – to display all data members.

(c) Answer the questions (i) and (ii) after going through the following class 2

class schoolbag

```

{   int pockets;
    public:
    schoolbag()           //Function 1
    { pockets=30;
    cout<<"The bag has pockets"<<endl;           }
    void company()       //Function 2
    {
    cout<<"The company of the Bag is ABC"<<endl;
    }
    schoolbag(int D)     //Function 3
    {
    pockets=D;
    cout<<"Now the Bag has pockets"<<pockets<<endl;
    }
}

```




Computer Science

```
~schoolbag() //Function 4
{ cout<<"Thanks"<<endl; }
};
```

(i) In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called ?

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

(d) Consider the following class state :

4

```
class State
{
protected:
int tp; //no. of tourist places
public:
State()
{ tp = 0; }
void inctp()
{ tp++; }
int gettp()
{ return tp; }
};
```

Write a code in C++ to publically derive another class 'District' with the following additional members derived in the Public visibility mode.

Data Members distname – char (50) population – long

Member functions :

- dinput() – To enter distname and population.
- doutput() – To display distname and population on screen.

3. (a) Write a function in C++ TWOTOONE() which accepts two array X[], Y[] and their size n as argument. Both the arrays X[] and Y[] have the same number of elements. Transfer the content from two arrays X[], Y[] to array Z[]. The even places (0,2,4...) of array Z[] should get the contents from the array X[] and odd places (1,3,5...) of array Z[] should get the contents from the array Y[].

3

Example : If the X[] array contains 30,60,90 and the Y[] array contains 10,20,50. Then Z[] should contain 30,10,60,20,90,50.

(b) Given an array A[10][12] whose base address is 10000. Calculate the memory location of A[2][5] if each element occupies 4 bytes and array is stored columnwise.

3

(c) Define member function delque() to perform delete operation on a linked queue where each node has the following structure :

4

```
struct node
{
char name[20]
int marks;
node *link;
};
class queue
{ node *front,*rear;
public:
queue() {front=rear=NULL; }
void delque();
};
```



(d) Write a function in C++ which accepts a 2D array of integers and its size arguments and displays the elements which lie on minor diagonal. [Top right to bottom left diagonal] 2
[Assuming the 2D array to be square matrix with odd dimension
i.e. 3 x 3, 5 x 5, 7x 7, etc ...]

For example
If the 2D array is

```
6 7 8
1 3 6
7 9 3
```

The following should be displayed :

```
8
3
7
```

(e) Evaluate the following postfix expression : 2
(show status of Stack after each operation)

100,40,8,/ ,20,10,- ,+ ,*

4.(a) A binary file "games.dat" contains data of 10 games where each game's data is an object of the following class : 1

```
class game
{
    int gameno;   char game_name[20];
    public:
    void enterdetails(){cin>>gameno; gets(game_name);}
    void enterdetails(){cout<<gameno<<endl<<game_name; };
```

With reference to this information, write C++ statement in the blank given below to move the file pointer to the end of file.

```
ifstream ifile;
game G;
ifile.open("games.dat",ios::binary|ios::in);

_____

cout<<ifile.tellg());
```

(b) Write a function Countaroma() to count and display the number of times "Aroma" occurs in a text file "Cook.txt". 2

Note : Only complete word "Aroma" should be counted. Words like "Aromatic" should not be counted.

(c) Given a binary file "SPORTS.DAT" containing records of the following class : 3

```
class Player
{ char PNO[10];      //player number
  char Name[20];    //Name of player
  int rank;         //rank of the player
  public:
  void EnterData()
  {gets(PNO);gets(Name);cin>>rank; }
  void DisplayData()
  { cout<<setw(12)<<PNO;
    cout<<setw(32)<<Name;
```



Computer Science

```
cout<<setw(3)<<rank<<endl;    }  
int Ret_rank() {return rank;}  };
```

Write a function in C++ that would read contents of the file "SPORTS.DAT" and display the details of those players whose rank is above 500.

5. (a) Observe the following table and answer the parts (i) and (ii) :

2

Table : Store

ItemCode	Item	Qty	Rate
10	Gel Pen Classic	1150	25
11	Sharpener	1500	10
12	Ball Pen 0.5	1600	12
13	Eraser	1600	5
15	Ball Pen 0.25	800	20

- (i) In the above table, can we have Qty as primary key. [Answer as yes/no]. Justify your answer.
(ii) What is the cardinality and degree of the above table ? Consider the following tables **SCHOOL** and **ADMIN** and answer
(iii)
(b) and (c) parts of this question :

Table :SCHOOL

CODE	TEACHERNAME	SUBJECT	DOJ	PERIODS	EXPERIENCE
1001	RAVI SHANKAR	ENGLISH	12/03/2000	24	10
1009	PRIYA RAI	PHYSICS	03/09/1998	26	12
1203	LISA ANAND	ENGLISH	09/04/2000	27	5
1045	YASHRAJ	MATHS	24/08/2000	24	15
1123	GANAN	PHYSICS	16/07/1999	28	3
1167	HARISH B	CHEMISTRY	19/10/1999	27	5
1215	UMESH	PHYSICS	11/05/1998	22	16

Table :ADMIN

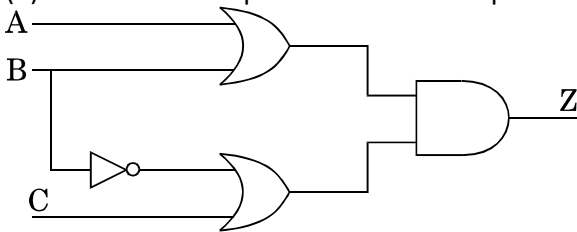
CODE	GENDER	DESIGNATION
1001	MALE	VICE PRINCIPAL
1009	FEMALE	COORDINATOR
1203	FEMALE	COORDINATOR
1045	MALE	HOD
1123	MALE	SENIOR TEACHER
1167	MALE	SENIOR TEACHER
1215	MALE	HOD

- (b) Write SQL statements for the following : 4
(i) To display TEACHERNAME, PERIODS of all teachers whose periods are more than 25.
(ii) To display all the information from the table SCHOOL in descending order of experience.
(iii) To display DESIGNATION without duplicate entries from the table ADMIN.
(iv) To display TEACHERNAME, CODE and corresponding DESIGNATION from tables SCHOOL and ADMIN of Male teachers.
(c) Give the output of the following SQL queries : 2
(i) SELECT DESIGNATION, COUNT (*) FROM ADMIN



Computer Science

- GROUP BY DESIGNATION HAVING COUNT (*) <2;
- (ii) SELECT max (EXPERIENCE) FROM SCHOOL;
- (iii) SELECT TEACHERNAME FROM SCHOOL
WHERE EXPERIENCE >12 ORDER BY TEACHERNAME;
- (iv) SELECT COUNT (*), GENDER FROM ADMIN GROUP BY GENDER;
6. (a) State Distributive law and verify it using truth table. 2
- (b) Write the equivalent Boolean Expression for the following Logic Circuit : 2



- (c) Convert the following Boolean expression into its equivalent Canonical Sum of Products form (SOP) : 1
- $$(U + V + W) (U + V + W') (U' + V + W) (U' + V' + W')$$
- (d) Reduce the following Boolean expression using K-Map : 3
- $$F(A,B,C,D) = \prod(0,1,2,4,5,6,8,10)$$
7. (a) Write any two differences between twisted pair and coaxial pair cable. 2
- (b) Define the following : 2
- (i) Firewall
- (ii) VoIP
- (c) Write any **two** examples of Server side Scripts. 1
- (d) What is cloud computing ? 1
- (e) Vidya Senior Secondary Public School in Nainital is setting up the network between its different wings. There are 4 wings named as SENIOR(S), JUNIOR(J), ADMIN(A) and HOSTEL(H). 4 Distance between various wings are given below :

Wing A to Wing S	100 m
Wing A to Wing J	200 m
Wing A to Wing H	400 m
Wing S to Wing J	300 m
Wing S to Wing H	100 m
Wing J to Wing H	450 m

Wing	Number of Computers
Wing A	20
Wing S	150
Wing J	50
Wing H	25

- (i) Suggest a suitable Topology for networking the computers of all wings.
- (ii) Name the most suitable wing where the Server should be installed. Justify your answer.
- (iii) Suggest where all should Hub(s)/Switch(es) be placed in the network.
- (iv) Which communication medium would you suggest to connect this school with its main branch in Delhi ?



AISSCE-2015
COMPUTER SCIENCE

Time allowed : 3 hours]

[Maximum marks : 70

Section – A

1. (a) Find the correct identifiers out of the following, which can be used for naming variable, constants or functions in a C++ program : 2

While, for, Float, new, 2ndName, A%B, Amount2, _Counter

(b) Observe the following program very carefully and write the names of those header file(s), which are essentially needed to compile and execute the following program successfully : 1

```
typedef char TEXT[80];  
void main()  
{  
    TEXT Str[] = "Peace is supreme";           int Index=0;  
    while (Str[Index]!='\0')  
        if (isupper(Str[Index]))  
            Str[Index++]='#';  
        else  
            Str[Index++]='*';  
    puts(Str);  
}
```

(c) Observe the following C++ code very carefully and rewrite it after removing any/all syntactical errors with each correction underlined. 2

Note : Assume all required header files are already being included in the program.

```
#Define float Max=70.0;  
Void main()  
{  
    int Speed           char Stop='N';           cin>>Speed;  
    if Speed>Max  
        Stop='Y';  
    cout<<Stop<<end;    }
```

(d) Write the output of the following C++ program code : 2

Note : Assume all required header files are already being included in the program.

```
void Position(int &C1,int C2=3)  
{  
    C1+=2;  
    C2+=Y;  
}  
void main()  
{  
    int P1=20, P2=4;  
    Position(P1);  
    cout<<P1<<" "<<P2<<end1;  
    Position(P2,P1);  
    cout<<P1<<" "<<P2<<end1;  
}
```

(e) Write the output of the following C++ program code : 3



Computer Science

Note : Assume all required header files are already being included in the program.

```
class Calc
{
    char Grade;          int Bonus;
public:
    Calc() { Grade='E'; Bonus=0;}
    void Down(int G)
    {   Grade--=G;   }
    void Up(int G)
    {   Grade+=G;    Bonus++;   }
    void Show()
    {   cout<<Grade<<"#"<<Bonus<<end1;   }   };
void main()
{
    Calc c;
    C.Down(2);
    C.Show();
    C.Up(7);
    C.Show();
    C.Down(2);
    C.Show();
}
```

(f) Study the following program and select the possible output(s) from the options (i) to (iv) following it. Also, write the maximum and the minimum values that can be assigned to the variable NUM. 2

Note : Assume all required header files are already being included in the program.

– random(n) function generates an integer between 0 and n – 1.

```
void main()
{
    randomize();          int NUM;
    NUM=random(3)+2;
    char TEXT[]="ABCDEFGHIIJK";
    for (int I=1;I<=NUM; I++)
    {
        for(int J=NUM; J<=7;J++)
        cout<<TEXT[J];
        cout<<end1;
    }
}
```

- | | | | |
|----------------------------------|-----------------------------------------|------------------------------------|-----------------------|
| (i) FGHI
FGHI
FGHI
FGHI | (ii) BCDEFGH
BCDEFGH
EFGH
EFGH | (iii) EFGH
EFGH
EFGH
EFGH | (iv) CDEFGH
CDEFGH |
|----------------------------------|-----------------------------------------|------------------------------------|-----------------------|

2. (a) What is a copy constructor ? Give a suitable example in C++ to illustrate with its definition within a class and a declaration of an object with the help of it. 2

(b) Observe the following C++ code and answer the questions (i) and (ii) :

```
class Traveller
{
    long PNR;          char TName[20];
public :
```



Computer Science

```

Traveller() //Function 1
{cout<<"Ready"<<end1;}
void Book(long P,char N[]) //Function 2
{ PNR = P; strcpy(TName, N); }
void Print() //Function 3
{ cout<<PNR << TName <<end1; }
~Traveller() //Function 4
{ cout<<"Booking cancelled!"<<end1; }
};

```

- (i) Fill in the blank statements in Line 1 and Line 2 to execute Function 2 and Function 3 respectively in the following code : 1

```

void main()
{
    Traveller T;
    _____ //Line 1
    _____ //Line 2
} //Stops here

```

- (ii) Which function will be executed at }//Stops here ? What is this function referred as ? 1
- (c) Write the definition of a class PIC in C++ with following description : 4

Private Members

- Pno //Data member for Picture Number (an integer)
- Category //Data member for Picture Category (a string)
- Location //Data member for Exhibition Location (a string)
- FixLocation //A member function to assign Exhibition Location as per category
//as shown in the following table

Category	Location
Classic	Amina
Modern	Jim Plaq
Antique	Ustad Khan

Public Members

- Enter() //A function to allow user to enter values Pno,
//category and call FixLocation() function
- SeeAll() //A function to display all the data members

- (d) Answer the questions (i) to (iv) based on the following : 4

```

class Exterior
{ int OrderId; char Address[20];
protected:
float Advance;
public:
Exterior(); void Book(); void View();
};
class Paint : public Exterior
{
int WallArea, ColorCode;
protected:

```

Computer Science

```

char Type;
public:
Paint();
void PBook();
void PView();
};
class Bill : public Paint
{
float Charges;
void Calculate();
public :
Bill();
void Billing();
void Print();
};

```

- (i) Which type of Inheritance out of the following is illustrated in the above example ?
 - Single Level Inheritance
 - Multi Level Inheritance
 - Multiple Inheritance
- (ii) Write the names of all the data members, which are directly accessible from the member functions of class Paint.
- (iii) Write the names of all the member functions, which are directly accessible from an object of class Bill.
- (iv) What will be the order of execution of the constructors, when an object of class Bill is declared ?

3. (a) Write the definition of a function Alter(int A[], int N) in C++, which should change all the multiples of 5 in the array to 5 and rest of the elements as 0. For example, if an array of 10 integers is as follows :

A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
55	43	20	16	39	90	83	40	48	25

After executing the function, the array content should be changed as follows :

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

- (b) A two dimensional array P[20] [50] is stored in the memory along the row with each of its element occupying 4 bytes, find the address of the element P[10] [30], if the element P[5] [5] is stored at the memory location 15000.
- (c) Write the definition of a member function Pop() in C++, to delete a book from a dynamic stack of TEXTBOOKS considering the following code is already included in the program.

```

struct TEXTBOOKS
{
char ISBN[20]; char TITLE[80];
TEXTBOOKS *Link;
};
class STACK

```



```

{
    TEXTBOOKS *Top;
public:
    STACK() { Top=NULL; }
    void Push();
    void Pop();
    ~STACK();
};

```

(d) Write a function REVCOL (int P[] [5], int N, int M) in C++ to display the content of a two dimensional array, with each column content in reverse order. 3

Note : Array may contain any number of rows.

For example, if the content of array is as follows :

15	12	56	45	51
13	91	92	87	63
11	23	61	46	81

The function should display output as :

```

11 23 61 46 81
13 91 92 87 63
15 12 56 45 51

```

(e) Convert the following infix expression to its equivalent Postfix expression, showing the stack contents for each step of conversion. 2

$$X / Y + U * (V - W)$$

4. (a) Write function definition for SUCCESS () in C++ to read the content of a text file STORY.TXT, count the presence of word STORY and display the number of occurrence of this word. 2

Note : The word STORY should be an independent word ignore type cases (i.e. lower/upper case)

Example :

If the content of the file STORY.TXT is as follows :

Success shows others that we can do it. It is possible to achieve success with hard work. Lot of money does not mean SUCCESS.

The function SUCCESS () should display the following :

3

(b) Write a definition for function Economic () in C++ to read each record of a binary file TEMS.DAT, find and display those items, which costs less than 2500. Assume that the file ITEMS.DAT is created with the help of objects of class ITEMS, which is defined below :

3

```

class ITEMS
{
    int ID; char GIFT[20]; float Cost;
public :
    void Get()
    { cin>>CODE; gets(GIFT); cin>>Cost; }
    void See()
    { cout<<ID<<"."<<GIFT<<"."<<Cost<<end1; }
    float GetCost() { return Cost; }.
};

```



(c) Find the output of the following C++ code considering that the binary file CLIENTS.DAT exists on the hard disk with records of 100 members. 1

```
class CLIENTS
{
    int Cno;char Name[20];
public :
        void In();        void Out();        };

void main()
{
    ifstream CF;
    CF.open("CLIENTS.DAT",ios::binary|ios::in);
    CLIENTS C;
    CF.read((char*) &C, sizeof(C));
    CF.read((char*) &C, sizeof(C));
    CF.read((char*) &C, sizeof(C));
    int POS=CF.tellg()/sizeof(C);
    cout<<"PRESENT RECORD:"<<POS<<endl;
    CF.close();        }
```

Section – C

5. (a) Observe the following table carefully and write the names of the most appropriate columns, which can be considered as (i) candidate keys and (ii) primary key. 2

Id	Product	Qty	Price	Transaction Date
101	Plastic Folder 12"	100	3400	2014-12-14
104	Pen Stand Standard	200	4500	2015-01-31
105	Stapler Medium	250	1200	2015-02-28
109	Punching Machine Big	200	1400	2015-03-12
103	Stapler Mini	100	1500	2015-02-02

(b) Consider the following DEPT and WORKER tables. Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii) : 6

Table : DEPT

DCODE	DEPARTMENT	CITY
D01	MEDIA	DELHI
D02	MARKETING	DELHI
D03	INFRASTRUCTURE	MUMBAI



Computer Science

D05	FINANCE	KOLKATA
D04	HUMAN RESOURCE	MUMBAI

Table : WORKER

WNO	NAME	DOJ	DOB	GENDER	DCODE
1001	George K	2013-09-02	1991-09-01	MALE	D01
1002	Ryma Sen	2012-12-11	1990-12-15	FEMALE	D03
1003	Mohitesh	2013-02-03	1987-09-04	MALE	D05
1007	Anil Jha	2014-01-17	1984-10-19	MALE	D04
1004	Manila Sahai	2012-12-09	1986-11-14	FEMALE	D01
1005	R SAHAY	2013-11-18	1987-03-31	MALE	D02
1006	Jaya Priya	2014-06-09	1985-06-23	FEMALE	D05

Note : DOJ refers to date of joining and DOB refers to date of Birth of workers.

- (i) To display Wno, Name, Gender from the table WORKER in descending order of Wno.
- (ii) To display the Name of all the FEMALE workers from the table WORKER.
- (iii) To display the Wno and Name of those workers from the table WORKER who are born between '1987-01-01' and '1991-12-01'.
- (iv) To count and display MALE workers who have joined after '1986-01-01'.
- (v) `SELECT COUNT(*), DCODE FROM WORKER
GROUP BY DCODE HAVING COUNT(*)>1;`
- (vi) `SELECT DISTINCT DEPARTMENT FROM DEPT;`
- (vii) `SELECT NAME, DEPARTMENT, CITY FROM WORKER W,DEPT D
WHERE W.DCODE=D.DCODE AND WNO<1003;`
- (viii) `SELECT MAX(DOJ), MIN(DOB) FROM WORKER;`

6. (a) Verify the following using Boolean Laws. 2

$$X + Y' = X.Y + X.Y' + X'.Y'$$

(b) Draw the Logic Circuit for the following Boolean Expression : 2

$$(U + V').W' + Z$$

(c) Derive a Canonical SOP expression for a Boolean function F, represented by the following truth table : 1

A	B	C	F(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0



Computer Science

1	1	1	1
---	---	---	---

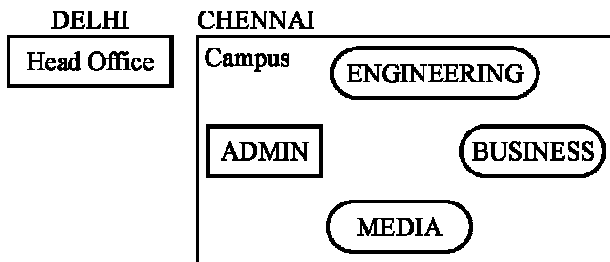
(d) Reduce the following Boolean Expression to its simplest form using K-Map : 3
 $F(X,Y,Z,W) = \sum(0,1,6,8,9,10,11,12,15)$

- 7.(a) Illustrate the layout for connecting 5 computers in a Bus and a Star topology of Networks. 1
 (b) What is a spam mail ? 1
 (c) Differentiate between ftp and http. 1
 (d) Out of the following, which is the fastest (i) wired and (ii) wireless medium of communication ? 1

Infrared, Coaxial Cable, Ethernet Cable, Microwave, Optical Fiber

- (e) What is Worm ? How is it removed ? 1
 (f) Out of the following, which all comes under cyber crime ? 1
 (i) Stealing away a brand new computer from a showroom.
 (ii) Getting in someone's social networking account without his consent and posting pictures on his behalf to harass him.
 (iii) Secretly copying files from server of a call center and selling it to the other organization.
 (iv) Viewing sites on an internet browser.
 (g) Perfect Edu Services Ltd. is an educational organization. It is planning to setup its India campus at Chennai with its head office at Delhi. The Chennai campus has 4 main buildings – ADMIN, ENGINEERING, BUSINESS and MEDIA.

You as a network expert have to suggest the best network related solutions for their problems raised in (i) to (iv), keeping in mind the distances between the buildings and other given parameters.



Shortest distances between various buildings :

ADMIN to ENGINEERING	55 m
ADMIN to BUSINESS	90 m
ADMIN to MEDIA	50 m
ENGINEERING to BUSINESS	55 m
ENGINEERING to MEDIA	50 m
BUSINESS to MEDIA	45 m

Computer Science

DELHI Head Office to CHENNAI Campus	2175 km
-------------------------------------	---------

Number of Computers installed at various buildings are as follows :

ADMIN	110
ENGINEERING	75
BUSINESS	40
MEDIA	12
DELHI Head Office	20

- i. Suggest the most appropriate location of the server inside the CHENNAI campus (out of the 4 buildings), to get the best connectivity for maximum no. of computers. Justify your answer.
1
 - ii. Suggest and draw the cable layout to efficiently connect various buildings within the CHENNAI campus for connecting the computers.
1
 - iii. Which hardware device will you suggest to be procured by the company to be installed to protect and control the internet uses within the campus ?
1
 - iv. Which of the following will you suggest to establish the online face-to-face communication between the people in the Admin Office of CHENNAI campus and DELHI Head Office ?
1
1. Cable TV
 2. Email
 3. Video Conferencing
 4. Text Chat



Computer Science

AISSCE- 2016 COMPUTER SCIENCE (083)

Time allowed : 3 hours

Maximum marks : 70

1. (a) Out of the following, find those identifiers, which cannot be used for naming Variable, Constants or Functions in a C++ program : 2

`_Cost, Price*Qty, float, Switch, Address One, Delete, Number12, do`

- (b) Jayapriya has started learning C++ and has typed the following program. When she compiled the following code written by her, she discovered that she needs to include some header files to successfully compile and execute it. Write the names of those header files, which are required to be included in the code. 1

```
void main()
{
    float A, Number, Outcome;
    cin>>A>>Number;
    Outcome=pow(A,Number);
    cout<<Outcome<<endl;
}
```

- (c) Rewrite the following C++ code after removing any/all syntactical errors with each correction underlined. 2

Note : Assume all required header files are already being included in the program.

```
#define Equation(p,q)=p+2*q
void main()
{
    float A=3.2;B=4.1;
    C=Equation(A,B);
    cout<<'Output='<<C<<endl;
}
```

- (d) Find and write the output of the following C++ program code : 2

Note : Assume all required header files are already included in the program.

```
typedef char STRING[80];
void MIXITNOW(STRING S)
{ int Size=strlen(S); for(int I=0;I<Size-1;I+=2)
{ char WS=S[I]; S[I]=S[I+1];
S[I+1]=WS;
}
for(I=1;I<Size;I+=2) if(S[I]>='M' && S[I]<='U')
S[I]='@';
}
void main()
{ STRING Word="CRACKAJACK"; MIXITNOW(Word); cout<<Word<<endl; }
```

- (e) Find and write the output of the following C++ program code : 3

Note : Assume all required header files are already being included in the program.

```
class Stock
{
    long int ID; float Rate; int Date;
public:
    Stock(){ID=1001;Rate=200;Date=1;}
    void RegCode(long int I,float R)
```



```

{
  ID=I;
  Rate=R;
}
void Change(int New,int DT)
{
  Rate+=New;
  Date=DT; }
void Show()
{
  cout<<"Date :"<<Date<<endl;
cout<<ID<<"#"<<Rate<<endl;
} };
void main()
{
  Stock A,B,C;
  A.RegCode(1024,150);
  B.RegCode(2015,300);
  B.Change(100,29);
  C.Change(-20,20);
  A.Show();
  B.Show();
  C.Show();
}
    
```

(f) Look at the following C++ code and find the possible output(s) from the options (i) to (iv) following it. Also, write the maximum and the minimum values that can be assigned to the variable CHANGER. 2

Note : Assume all the required header files are already being included in the code.

- The function random(n) generates an integer between 0 and n – 1
- void main()


```

{
  randomize();
  int CHANGER;
  CHANGER=random(3);
  char CITY[][25]={"DELHI","MUMBAI","KOLKATA","CHENNAI"};
  for(int I=0;I<=CHANGER;I++)
  {
  for(int J=0;J<=I;J++)  cout<<CITY[J];  cout<<endl;      }      }
      
```

(i)	(ii)
DELHI DELHIMUMBAI DELHIMUMBAIKOLKATA	DELHI DELHIMUMBAI DELHIMUMBAIKOLKATA DELHIMUMBAIKOLKATACHENNAI



Computer Science

(iii)	(iv)
MUMBAI	KOLKATA
MUMBAIKOLKATA	KOLKATACHENNAI
MUMBAIKOLKATACHENNAI	

2. (a) Differentiate between Constructor and Destructor functions giving suitable example using a class in C++. When does each of them execute ? **2**

(b) Observe the following C++ code and answer the questions (i) and (ii). Assume all necessary files are included :

```
class FICTION
{
long FCode; char FTitle[20]; float FPrice; Public:
FICTION() //Member Function 1
{ cout<<"Bought"<<endl;
  FCode=100;strcpy(FTitle,"Noname");FPrice=50;
}
FICTION(int C,char T[],float P) // Member Function 2
{ FCode=C;
  strcpy(FTitle,T);
  FPrice=P;
}
void Increase(float P) // Member Function 3
{ FPrice+=P; }
void Show() // Member Function 4
{ cout<<FCode<<":"<<FTitle<<":"<<FPrice<<endl; }
~FICTION() // Member Function 5
{ cout<<"Fiction removed!" <<endl; }
};

void main() //Line 1
{ //Line 2
  FICTION F1,F2(101,"Dare",75); //Line 3 for (int l=0;l<4;l++) //Line 4
  { //Line 5
    F1.Increase(20);F2.Increase(15); //Line 6
    F1.Show();F2.Show(); //Line 7
  } //Line 8
} //Line 9
```

(i) Which specific concept of object oriented programming out of the following is illustrated by Member Function 1 and Member Function 2 combined together ? **1**

- Data Encapsulation
- Data Hiding
- Polymorphism
- Inheritance

(ii) How many times the message "Fiction removed!" will be displayed after executing the above C++ code ? Out of Line 1 to Line 9, which line is responsible to display the message "Fiction removed!" ? **1**

(c) Write the definition of a class METROPOLIS in C++ with following description : **4**

Private Members

- MCode //Data member for Code (an integer)
- MName //Data member for Name (a string)
- MPop //Data member for Population (a long int)
- Area //Data member for Area Coverage (a float)
- PopDens //Data member for Population Density (a float)
- CalDen() //A member function to calculate Density as PopDens/Area

Public Members

- Enter() //A function to allow user to enter values of
// Mcode,MName,MPop,Area and call CalDen() function
- ViewALL() //A function to display all the data members also display a message
//“Highly Populated Area” if the Density is more than 12000

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

4

```
class PRODUCT
{ int Code; char Item[20];
protected:
float Qty;
public:
PRODUCT();
void GetIn(); void Show();          };
class WHOLESALER
{ int WCode;
protected:
char Manager[20];
public:
WHOLESALER();
void Enter();
void Display();          };
class SHOWROOM : public PRODUCT, private WHOLESALER
{ char Name[20],City[20];
public:
SHOWROOM();
void Input();
void View();          };
```

- (i) Which type of Inheritance out of the following is illustrated in the above example ?
- Single Level Inheritance
 - Multi Level Inheritance
 - Multiple Inheritance
- (ii) Write the names of all the data members, which are directly accessible from the member functions of class SHOWROOM.
- (iii) Write the names of all the member functions, which are directly accessible by an object of class SHOWROOM.
- (iv) What will be the order of execution of the constructors, when an object of class SHOWROOM is declared ?
3. (a) Write the definition of a function FixPay(float Pay[], int N) in C++, which should modify each element of the array Pay having N elements, as per the following rules : **2**



Computer Science

Existing Value of Pay	Pay to be changed to
If less than 100000	Add 25% in the existing value
If >=100000 and <20000	Add 20% in the existing value
If >=200000	Add 15% in the existing value

(b) T[20][50] is a two dimensional array, which is stored in the memory along the row with each of its element occupying 4 bytes, find the address of the element T[15][5], if the element T[10][8] is stored at the memory location 52000. **3**

(c) Write the definition of a member function INSERT() for a class QUEUE in C++, to insert an ITEM in a dynamically allocated Queue of items considering the following code is already written as a part of the program. **4**

```
struct ITEM
{
    int INO; char INAME[20];
    ITEM *Link; };
class QUEUE
{ ITEM *R,*F; public :
    QUEUE() {R=NULL;F=NULL;} void INSERT(); void DELETE();
    ~QUEUE(); };
```

(d) Write definition for a function SHOWMID(int P[][5], int R, int C) in C++ to display the elements of middle row and middle column from a two dimensional array P having R number of rows and C number of columns.

For example, if the content of array is as follows : **3**

115	112	116	101	125
103	101	121	102	101
185	109	109	160	172

The function should display the following as output :

103 101 121 102 101 116 121 109

(e) Convert the following Infix expression to its equivalent Postfix expression, showing the stack contents for each step of conversion. **2**

A/(B+C)*D-E

4. (a) Write function definition for WORD4CHAR() in C++ to read the content of a text file FUN.TXT, and display all those words, which has four characters in it. **2**

Example :

If the content of the file fun.TXT is as follows :

When I was a small child, I used to play in the garden with my grand mom. Those days were amazingly funful and I remember all the moments of that time

The function WORD4CHAR() should display the following :

When used play with days were that time

(b) Write a definition for function BUMPER() in C++ to read each object of a binary file GIFTS.DAT, find and display details of those gifts, which has remarks as "ÖN DISCOUNT". Assume that the file GIFTS.DAT is created with the help of objects of class GIFTS, which is defined below : **3**

```
class GIFTS
{
```



```
int ID;char Gift[20],Remarks[20]; float Price;
public:
void Takeonstock()
{ cin>>ID;gets(Gift);gets(Remarks);cin>>Price; }
void See()
{ cout<<ID<<"."<<Gift<<"."<<Price<<"."<<Remarks<<endl; }
char *GetRemarks() {return Remarks;}    };
```

(c) Find the output of the following C++ code considering that the binary file MEM.DAT exists on the hard disk with a data of 1000 members : 1

```
class MEMBER
{
int Mcode;char MName[20]; public:
void Register();void Display();    };
void main()
{
fstream MFile;
MFile.open("MEM.DAT", ios::binary|ios::in);
MEMBER M;
MFile.read((char*)&M, sizeof(M)); cout<<"Rec:"<<MFile.tellg()/sizeof(M)<<endl;
MFile.read((char*)&M, sizeof(M)); MFile.read((char*)&M, sizeof(M));
cout<<"Rec:"<<MFile.tellg()/sizeof(M)<<endl;
MFile.close();    }
```

5(a) Observe the following STUDENTS and EVENTS tables carefully and write the name of the RDBMS operation which will be used to produce the output as shown in LIST. Also, find the Degree and Cardinality of the LIST. 2

STUDENTS

NO	NAME
1	Tara Mani
2	Jaya Sarkar
3	Tarini Trikha

EVENTS

EVENTCODE	EVENTNAME
1001	Programming
1002	IT Quiz

LIST

NO	NAME	EVENTCODE	EVENTNAME
1	Tara Mani	1001	Programming
1	Tara Mani	1002	IT Quiz
2	Jaya Sarkar	1001	Programming
2	Jaya Sarkar	1002	IT Quiz
3	Tarini Trikha	1001	Programming
3	Tarini Trikha	1002	IT Quiz

(b) Write SQL queries for (i) to (iv) and find outputs for SQL queries (v) to (viii), which are based on the tables. 6

Table : VEHICLE

CODE	VTYPE	PERKM
101	VOLVO BUS	160
102	AC DELUXE BUS	150

Computer Science

103	ORDINARY BUS	90
105	SUV	40
104	CAR	20

Note : PERKM is Freight Charges per kilometer , VTYPE is Vehicle Type

Table : TRAVEL

NO	NAME	TDATE	KM	CODE	NOP
101	Janish Kin	2015-11-13	200	101	32
103	Vedika Sahai	2016-04-21	100	103	45
105	Tarun Ram	2016-03-23	350	102	42
102	John Fen	2016-02-13	90	102	40
107	Ahmed Khan	2015-01-10	75	104	2
104	Raveena	2016-05-28	80	105	4
106	Kripal Anya	2016-02-06	200	101	25

Note :

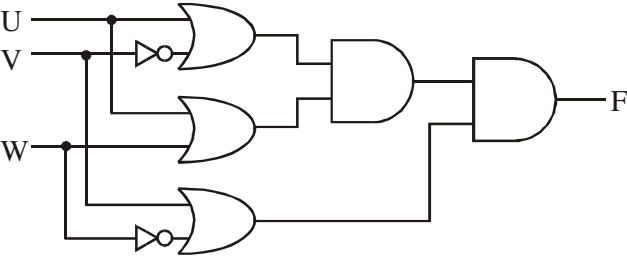
- NO is Traveller Number
- KM is Kilometer travelled
- NOP is number of travellers travelled in vehicle
- TDATE is Travel Date

- (i) To display NO, NAME, TDATE from the table TRAVEL in descending order of NO.
- (ii) To display the NAME of all the travellers from the table TRAVEL who are travelling by vehicle with code 101 or 102.
- (iii) To display the NO and NAME of those travellers from the table TRAVEL who travelled between '2015-12-31' and '2015-04-01'.
- (iv) To display all the details from table TRAVEL for the travellers, who have travelled distance more than 100 KM in ascending order of NOP.
- (v) SELECT COUNT (*), CODE FROM TRAVEL GROUP BY CODE HAVING COUNT(*)>1;
- (vi) SELECT DISTINCT CODE FROM TRAVEL;
- (vii) SELECT A.CODE,NAME,VTYPE FROM TRAVEL A,VEHICLE B WHERE A.CODE=B.CODE AND KM<90;
- (viii) SELECT NAME,KM*PERKM FROM TRAVEL A, VEHICLE B WHERE A.CODE=B.CODE AND A.CODE='105';

6. (a) Verify the following using Boolean Laws : 2

$$A' + B'.C = A'.B'.C' + A'.B.C' + A'.B.C + A'.B'.C + A.B'.C$$

(b) Write the Boolean Expression for the result of the Logic Circuit as shown below : 2





Computer Science

- (c) Derive a Canonical POS expression for a Boolean function F, represented by the following truth table : 1

P	Q	R	F(P, Q, R)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- (d) Reduce the following Boolean Expression to its simplest form using K-Map : 3
 $F(X,Y,Z,W) = \Sigma(2,6,7,8,9,10,11,13,14,15)$

7.(a) Give two examples of PAN and LAN type of networks.

(b) Which protocol helps us to browse through web pages using internet browsers ? Name any one internet browser. 1

(c) Write two advantages of 4G over 3G Mobile Telecommunication Technologies in terms of speed and services. 1

(d) Write two characteristics of Web 2.0. 1

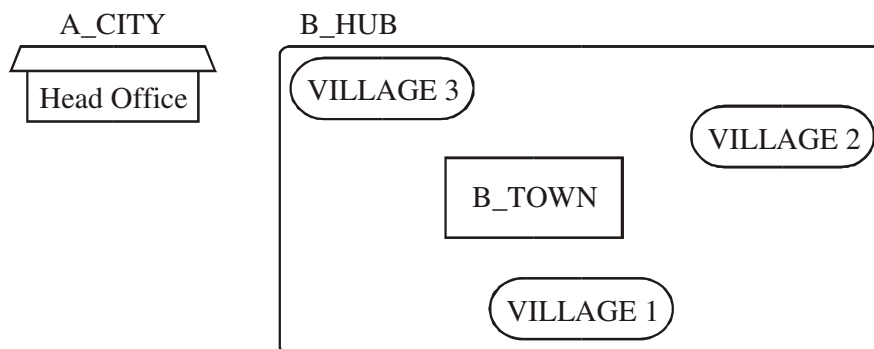
(e) What is the basic difference between Trojan Horse and Computer Worm ? 1

(e) Categorise the following under Client Side and Server Side script category : 1

- (i) VB Script
- (ii) ASP
- (iii) JSP
- (iv) JavaScript

(f) Uplifting Skills Hub India is a knowledge and skill community which has an aim to uplift the standard of knowledge and skills in the society. It is planning to setup its training centers in multiple towns and villages pan India with its head offices in the nearest cities. They have created a model of their network with a city, a town and 3 villages as follows.

As a network consultant, you have to suggest the best network related solutions for their issues/problems raised in (i) to (iv) keeping in mind the distances between various locations and other given parameters.



Shortest distances between various locations :

Computer Science

VILLAGE 1 to B_TOWN	2 KM
VILLAGE 2 to B_TOWN	1.0 KM
VILLAGE 3 to B_TOWN	1.5 KM
VILLAGE 1 to VILLAGE 2	3.5 KM
VILLAGE 1 to VILLAGE 3	4.5 KM
VILLAGE 2 to VILLAGE 3	2.5 KM
A_CITY Head Office to B_HUB	25 KM

Number of Computers installed at various locations are as follows :

B_TOWN	120
VILLAGE 1	15
VILLAGE 2	10
VILLAGE 3	15
A_CITY OFFICE	6

Note :

- In Villages, there are community centers, in which one room has been given as training center to this organization to install computers.
 - The organization has got financial support from the government and top IT companies.
- (i) Suggest the most appropriate location of the SERVER in the B_HUB (out of the 4 locations), to get the best and effective connectivity. Justify your answer. **1**
- (ii) Suggest the best wired medium and draw the cable layout (location to location) to efficiently connect various locations within the B_HUB. **1**
- (iii) Which hardware device will you suggest to connect all the computers within each location of B_HUB ? **1**
- (iv) Which service/protocol will be most helpful to conduct live interactions of Experts from Head Office and people at all locations of B_HUB ? **1**



Marking Scheme (Solution) of AISSCE-2016 Question Paper

- 1 (a) Price*Qty float Address One do 2
- (b)
 - iostream.h OR iomanip.h
 - math.h1
- (c) #define Equation(p,q) p+2*q void main() 2
 {
 float A=3.2, B=4.1;
 float C=Equation(A,B);
 cout<<"Output="<<C<<endl;
 }
- (d) RCCAAKAJKC 2
- (e) Find and write the output of the following C++ program code: 3
 Note: Assume all required header files are already being included in the program.
- ```
class Stock
{
long int ID;
```
- (e) Date :1 3  
 1024#150  
 Date :29  
 2015#400  
 Date :20  
 1001#180
- (f) (i) 2  
 DELHI  
 DELHIMUMBAI  
 DELHIMUMBAIKOLKATA

*Minimum Value of CHANGER = 0 Maximum Value of CHANGER = 2*

- 2 (a) Example: 2

| Constructor                                       | Destructor                                                            |
|---------------------------------------------------|-----------------------------------------------------------------------|
| A constructor function has same name as the class | A destructor function has same name as the class preceded by ~ symbol |

```
class Exam
{
int Eno; float Marks; public:
Exam() //Constructor
{
Eno=1; Marks = 100;
cout<<"Constructor
executed..."<<endl;
}
```

```
void Show()
{ cout<<Eno<<"#"<<Marks<<endl; }
~Exam() //Destructor
{
cout<<"Exam Over"<<endl; } };
void main()
{ Exam E; //Executes constructor
E.Show();
```

|   |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |   |
|---|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|   | b.(i)     | Polymorphism                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 1 |
|   | b. (ii)   | 2 times<br>Line 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 1 |
|   | (c)       | <pre> class METROPOLIS {     int Mcode;     char MName[20];     long int MPop;     float Area;     float PopDens; void CalDen(); public:     void Enter();     void ViewALL(); }; void METROPOLIS::Enter() {     cin&gt;&gt;Mcode;     gets(MName); //OR cin&gt;&gt;MName;     cin&gt;&gt;MPop;     cin&gt;&gt;Area;     CalDen(); } void METROPOLIS::ViewALL() {     cout&lt;&lt;Mcode&lt;&lt;MName&lt;&lt;MPop&lt;&lt;Area&lt;&lt;PopDens; //Ignore endl     if(PopDens&gt;12000)         cout&lt;&lt;"Highly Populated Area";    //Ignore endl } void METROPOLIS::CalDen() {     PopDens= PopDens/Area; //OR PopDens = MPop/Area }                     </pre> | 4 |
|   | (d) (i)   | Multiple Inheritance                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 1 |
|   | (d) (ii)  | Name, City, Manager, Qty                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 1 |
|   | (d) (iii) | Input(), View(), GetIn(), Show()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 1 |
|   | (d) (iv)  | <b>(i)</b> PRODUCT()<br><b>(ii)</b> WHOLESALER()<br><b>(iii)</b> SHOWROOM()                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 1 |
| 3 | (a)       | <pre> void FixPay(float Pay[ ], int N) {     for (int i=0;i&lt;N;i++)         if(Pay[i]&lt;100000)             Pay[i]+= 0.25 * Pay[i];             else if (Pay[i]&gt;=100000 &amp;&amp; Pay[i]&lt;20000)                 Pay[i]+= 0.2 * Pay[i];             else if(Pay[i]&gt;=200000)                 Pay[i]+= 0.15 * Pay[i]; }                     </pre>                                                                                                                                                                                                                                                                                                     | 2 |



(b)  $Loc(T[I][J])$   
 $= \text{BaseAddress} + W [(I - LBR) * C + (J - LBC)]$   
 (where  
 $W = \text{size of each element} = 4 \text{ bytes,}$   
 $R = \text{Number of Rows} = 20, C = \text{Number of Columns} = 50)$   
 Assuming  $LBR = LBC = 0$   
 $LOC(T[10][8])$   
 $52000 = \text{BaseAddress} + W [I * C + J]$   
 $52000 = \text{BaseAddress} + 4 [10 * 50 + 8]$   
 $52000 = \text{BaseAddress} + 4 [500 + 8]$   
 $52000 = \text{BaseAddress} + 4 \times 508$   
 $\text{BaseAddress} = 52000 - 2032$   
 $= 49968$   
 $LOC(T[15][5]) = \text{BaseAddress} + W [I * C + J]$   
 $= 49968 + 4 [15 * 50 + 5]$   
 $= 49968 + 4 [750 + 5]$   
 $= 49968 + 4 \times 755$   
 $= 49968 + 3020$   
 $= 52988 \text{ OR}$   
 $Loc(T[I][J])$   
 $= \text{ReferenceAddress} + W [(I - LR) * C + (J - LC)]$  (where

$W = \text{size of each element} = 4 \text{ bytes,}$   
 $R = \text{Number of Rows} = 20, C = \text{Number of Columns} = 50)$   
 $\text{ReferenceAddress} = \text{Address of given cell } T[10][8] = 52000$   
 $LR = \text{Row value of given cell} = 10$   
 $LC = \text{Column value of given cell} = 8$   
 $LOC(T[15][5]) = LOC(T[10][8]) + 4 [(15 - 10) * 50 + (5 - 8)]$   
 $LOC(T[15][5]) = 52000 + 4 [5 * 50 + (-3)]$   
 $= 52000 + 4 [250 - 3]$   
 $= 52000 + 4 \times 247$   
 $= 52000 + 988$   
 $= 52988$

(c) `void QUEUE::INSERT() {`

```

ITEM *T = new ITEM;
cin>>T->INO;
gets(T->INAME); //OR cin>> T->INAME;
T->Link = NULL;
if(R==NULL)
{
 F=T; R=T;
}
else
{
 R->Link=T; R=T;
} }

```

(d) void SHOWMID(int P[][5],int R,int C)

```

{
 for (int J=0;J<C;J++)
 cout<<P[R/2][J]<< " ";
 cout<<endl;
 for (int l=0;l<R;l++)

```

3

```

 cout<<P[l][C/2]<< " ";
 }
OR void SHOWMID(int P[][5],int R,int C)
{
 if(R%2!=0)
 {
 for (int J=0;J<C;J++)
 cout<<P[R/2][J]<< " ";
 }
 else
 cout<<"No Middle Row";
 cout<<endl;
 if(C%2!=0)
 {
 for (int l=0;l<R;l++)
 cout<<P[l][C/2]<< " ";
 }
 else
 cout<<"No Middle Column";
}

```

(e)  $A/(B+C)*D-E$   
 $= (((A / (B+C)) * D) - E)$

2

| Element | Stack of Operators | Postfix Expression |
|---------|--------------------|--------------------|
| (       |                    |                    |

|   |    |                |
|---|----|----------------|
| ( |    |                |
| ( |    |                |
| A |    | A              |
| / | /  | A              |
| ( | /  | A              |
| B | /  | AB             |
| + | /+ | AB             |
| C | /+ | ABC            |
| ) | /  | ABC+           |
| ) |    | ABC+ /         |
| * | *  | ABC+ /         |
| D | *  | ABC+ / D       |
| ) |    | ABC+ / D *     |
| - | -  | ABC+ / D *     |
| E | -  | ABC+ / D * E   |
| ) |    | ABC+ / D * E - |

= ABC+ / D \* E -

OR

A / (B+C) \* D - E

= (A / (B+C) \* D - E)

| Element | Stack of Operators | Postfix Expression |
|---------|--------------------|--------------------|
| (       | (                  |                    |
| A       | (                  | A                  |
| /       | (/                 | A                  |
| (       | ((                 | A                  |
| B       | ((                 | AB                 |
| +       | ((+                | AB                 |
| C       | ((+                | ABC                |
| )       | (/                 | ABC+               |
| *       | (*                 | ABC+ /             |
| D       | (*                 | ABC+ / D           |
| -       | (-                 | ABC+ / D *         |
| E       | (-                 | ABC+ / D * E       |
| )       |                    | ABC+ / D * E -     |

= ABC+ / D \* E -

4 (a)

```
void WORD4CHAR()
{
 ifstream Fil;
 Fil.open("FUN.TXT");
 char W[20];
 Fil>>W;
 while(!Fil.eof()) //OR while(Fil)
 {
 if (strlen(W) == 4) //Ignore words ending with '.'
 cout<<W<< " ";
 Fil>>W;
 }
 Fil.close(); //Ignore
}
```

2

(b) void BUMPER()

```
{
 GIFTS G;
 ifstream fin;
 fin.open("GIFTS.DAT", ios::binary);
 while(fin.read((char*)&G, sizeof(G)))
 {
 if(strcmp(G.GetRemarks(),"ON DISCOUNT")==0)
 G.See();
 }
 fin.close(); //Ignore
}
```

3

(c) Rec:1

Rec:3

|   |           |                                                                                                                                                                                                                                                                                                                                                                            |   |
|---|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| 5 | (a)       | Cartesian Product<br><br>Degree = 4<br>Cardinality = 6                                                                                                                                                                                                                                                                                                                     | 2 |
|   | (b) (i)   | SELECT NO, NAME, TDATE FROM TRAVEL<br>ORDER BY NO DESC;                                                                                                                                                                                                                                                                                                                    | 1 |
|   | (b) (ii)  | SELECT NAME FROM TRAVEL<br>WHERE CODE='101' OR CODE='102';<br>OR<br>SELECT NAME FROM TRAVEL<br>WHERE CODE=101 OR CODE=102;<br><br>OR<br>SELECT NAME FROM TRAVEL<br>WHERE CODE IN ('101','102'); OR<br>SELECT NAME FROM TRAVEL WHERE CODE IN (101,102);                                                                                                                     | 1 |
|   | (b) (iii) | SELECT NO, NAME from TRAVEL<br>WHERE TDATE >= '2015-04-01' AND TDATE <= '2015-12-31';<br>OR<br>SELECT NO, NAME from TRAVEL<br>WHERE TDATE BETWEEN '2015-04-01' AND '2015-12-31';<br>OR<br>SELECT NO, NAME from TRAVEL<br>WHERE TDATE <= '2015-12-31' AND TDATE >= '2015-04-01';<br>OR<br>SELECT NO, NAME from TRAVEL<br>WHERE TDATE BETWEEN '2015-12-31' AND '2015-04-01'; | 1 |
|   | (b) (iv)  | SELECT * FROM TRAVEL<br>WHERE KM > 100 ORDER BY NOP;                                                                                                                                                                                                                                                                                                                       | 1 |
|   | (v)       | SELECT COUNT(*),CODE FROM TRAVEL<br>GROUP BY CODE HAVING COUNT(*)>1;                                                                                                                                                                                                                                                                                                       | 1 |
|   | (b) (v)   | <u>COUNT(*)CODE</u><br>2      101<br>2      102                                                                                                                                                                                                                                                                                                                            | ½ |
|   | (b) (vi)  | <u>DISTINCT CODE</u><br>101<br>102<br>103<br>104<br>105                                                                                                                                                                                                                                                                                                                    | ½ |



- (b) (vii) CODENAMEVTYPE  
**104** Ahmed Khan CAR  
**105** Raveena SUV

½

- (b) (viii) NAMEKM\*PERKM  
 Raveena 3200

½

- 6 (a) LHS 2

$$\begin{aligned}
 &A' + B'.C \\
 &= A'.(B + B').(C + C') + (A + A').B'.C \\
 &= A'.B.C + A'.B.C' + A'.B'.C + A'.B'.C' + A.B'.C + A'.B'.C \\
 &= A'.B.C + A'.B.C' + A'.B'.C + A'.B'.C' + A.B'.C \\
 &= A'.B'.C' + A'.B.C' + A'.B.C + A'.B'.C + A.B'.C \\
 &= RHS \\
 &OR \\
 &RHS = A'.B'.C' + A'.B.C' + A'.B.C + A'.B'.C + A.B'.C \\
 &= A'.B'.C + A'.B'.C' + A'.B.C + A'.B.C' + A.B'.C \\
 &= A'.B'.(C+C') + A'.B.(C+C') + A.B'.C \\
 &= A'.B' + A'.B + A.B'.C \\
 &= A'.(B'+B) + A.B'.C \\
 &= A' + A.B'.C \\
 &= (A' + A).(A' + B'.C) \\
 &= A' + B'.C = LHS
 \end{aligned}$$

- (b)  $((U + V').(U + W)).(V + W')$  2  
 OR  
 $(U + V').(U + W).(V + W')$

- (c)  $F(P,Q,R)=(P+Q+R).(P+Q'+R').(P'+Q+R).(P'+Q+R')$  1  
 OR  
 $F(P,Q,R)=\pi(0,3,4,5)$

- (d) 3

|      | X'Y' | X'Y | XY | XY' |
|------|------|-----|----|-----|
| Z'W' | 0    | 4   | 12 | 1   |
| Z'W  | 1    | 5   | 13 | 9   |
| ZW   | 1    | 7   | 15 | 1'  |
| ZW'  | 1    | 8   | 14 | 10  |

OR

$$\begin{aligned}
 &F(X,Y,Z, \\
 &W) = \\
 &XY' + \\
 &ZW' + \\
 &XW + \\
 &YZ
 \end{aligned}$$

|      | Z'W' | Z'W | ZW | ZW' |
|------|------|-----|----|-----|
| X'Y' | 0    | 1   | 3  | 2   |
| X'Y  | 4    | 5   | 7  | 6   |
| XY   | 12   | 1   | 14 | 16  |
| XY'  | 1    | 9   | 11 | 10  |



## Computer Science

7 (a) 1

| PAN Examples                                | LAN Examples                      |
|---------------------------------------------|-----------------------------------|
| Connecting two cell phones to transfer data | Connecting computers in a school  |
| Connecting smartphone to a smart watch      | Connecting computers in an office |

Note: Any one example of each

OR

Any other one/two correct examples for each of PAN and LAN

(b) Protocol: HTTP OR TCP/IP 1

Browser: Chrome OR Internet Explorer OR Firefox OR OPERA OR SAFARI OR any other correct Browser Name

(c) 1

| 4G                           | 3G                            |
|------------------------------|-------------------------------|
| Speed approximately 100 mbps | Speed approximately 2 mbps    |
| LTE True mobile broadband    | Data services with multimedia |

OR

Any other two correct advantages of 4G over 3G in terms of speed and services

(d) 1

- Makes web more interactive through online social media
- Supports easy online information exchange
- Interoperability on the internet
- Video sharing possible in the websites

OR

Any two of the above or any other two correct characteristics of Web 2.0

(e) 1

Trojan Horse

It is a "Malware" computer program presented as useful or harmless in order to induce the user to install and run them.

Computer Worm

It is a self-replicating computer program. It uses a network to send copies of itself to other nodes (computers on the network) and it may do so without any user intervention.

OR Any other correct difference between Trojan Horse and Computer Worm

(f) 1

Client Side Scripts

VB Script

Java Script

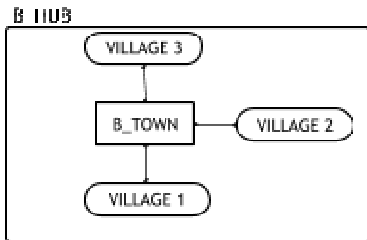
Server Side Scripts

ASP

JSP

(g) (i) B\_TOWN. Since it has the maximum number of computers and is closest to all other locations. 1

(g) (ii) Best Wired Medium : Optical Fibre 1



(g) (iii) Switch OR Hub 1

(g) (iv) Videoconferencing OR VoIP OR any other correct service/protocol 1



## **PREPARATION FOR EXAMINATION**

1. Write down how many exams you have and the days on which you have to sit them. Then organize your study accordingly.
2. Organize your study space and make sure you feel as comfortable and able to focus as possible.
3. Use flowcharts and Diagrams and condense your revision notes into one-page diagrams.
4. One of the most effective ways to prepare for exams is to practice on old papers.
5. Explain your answers with peer group and clear your doubts with the help of teachers.
6. Students should let go off the cramming part as it is not a good method of studying. Instead of this, they should make short notes while studying.
7. Studies have shown that for long-term retention of knowledge, taking regular breaks really helps.
8. Plan your exam well in advance and make sure you get everything ready well in advance of the exam - don't leave it to the day before to suddenly realize you don't know the way, or what you're supposed to bring. Check all the rules and requirements, and plan your route and journey time. If possible, do a test run of the trip; if not, write down clear directions.
9. What you eat can really have an impact on energy levels and focus. Keep your body and brain well-fueled by choosing nutritious foods. Drink a lot of water.
10. When students write neatly and present their answer sheet in a proper way, they have more chances of securing high marks.