

K.S.R. COLLEGE OF ENGINEERING: TIRUCHENGODE – 637 215.
(An Autonomous Institution, Affiliated to Anna University, Chennai)
DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGINEERING

COURSE / LESSON PLAN SCHEDULE (2019 – 2020)

FACULTY NAME(s) : A. VELLIANGIRI

CLASS: II B.E (ECE)

COURSE CODE / NAME : 18CS332 C++ and Data Structures

A). REFERENCE BOOKS:

1. Robert Lafore, "Object oriented programming in C++", Galgotia Publication, 4th Edition, 2014.
2. B. Trivedi, Programming with ANSI C++, Oxford University Press 2007.
3. A.K. Sharma, Data Structures using C, 1st Edition, Pearson Education, 2011.
4. R.F. Gilberg, B.A. Forouzan, Data Structures, 2nd Edition, Thomson India 2005.

B). TEXT BOOKS

1. E. Balagurusamy, Object Oriented Programming with C++, 6th Edition, Tata McGraw Hill India Pvt. Ltd., Newdelhi, 2013.
2. M. A. Weiss, Data Structures and Algorithm Analysis in C, 2nd Edition, Pearson Education 2015.

C). EXTRA BOOK

1. Debasis Samanta, "Classic Data Structures", Second Edition, PHI – 2009.

D). LEGEND:

L - Lecture

T - Tutorial

Rx - Reference Book

Ex – Extra Book

TX – Text Book

BB - Black Board

LCD – Liquid Crystal Display

Sl. No	Lecture Hour	Topics to be covered	Teaching Aid Required	Book No./Page No
UNIT – I INTRODUCTION TO C++				
1	L01	Object Oriented Programming Concepts	BB	Tx1/pp 6-10,Rx1/pp 14-19 Rx2/pp 5-26
2	L02	Difference between object oriented and procedure oriented.	BB	Tx1/pp 4-6,Rx1/pp 19-20 Rx2/pp 27-34
3		Benefits – Applications.	BB	Tx1/pp 10-13,Rx2/pp 35
4	L03	Introduction to C++ - Tokens.	BB	Tx1/pp 16-29,Rx2/pp 39
5	L04	Data Types - Operators.	BB	Tx1/pp 29-49,Rx2/pp 40-41 & pp 58-65
6	L05 & L06	Classes- Objects.	BB	Tx1/pp 88-96,Rx2/pp 69-104
7	L07	Default Arguments.	BB	Tx1/pp 75-77,Rx2/pp 115-123
8	L08	Static member functions - Static data members.	BB	Tx1/pp 104-108,Rx2/pp 149-151
9	L09	Friend functions. Constant Member Functions	BB	Tx1/pp 113-118,Rx2/pp 140-147, Tx1/pp 120
UNIT – II CONSTRUCTORS AND INHERITANCE				
10	L10	Constructors	BB	Tx1/pp 129-144,Rx2/pp 177-215
11	L11			
12	L12	Destructors	BB	Tx1/pp 144-147,Rx2/pp 216-223
13	L13	Unary Operator overloading	BB	Tx1/pp 152-155,Rx1/pp 291-298 Rx2/pp 236-239
14	L14	Binary Operator overloading	BB	Tx1/pp 155-161,Rx1/pp 299-302 Rx2/pp 240-242
15	L15	Inheritance	BB	Tx1/pp 179-204,Rx2/pp 362-384
16	L16			
17	L17	Virtual functions-Pure Virtual Function	BB	Tx1/pp 243-249,Rx1/pp 453-462

				Rx2/pp 410-431
18	L18	Exception Handling Class Templates	LCD	Tx1/pp 340-356,Rx2/pp 328-331 Tx1/pp 319-324.
UNIT – III DATA STRUCTURES & ALGORITHMS				
19	L19	Analysis of Algorithms.	BB	Tx2/pp 31-50,Rx3/pp 73-78
20	L20	Abstract Data types – Lists.	BB	Tx2/pp 57-78,Rx3/pp 207-243
21	L21	Stacks	BB / LCD	Tx2/pp 78-95,Rx3/pp 131-145
22	L22			
23	L23	Queues	BB	Tx2/pp 95-100,Rx3/pp 150-156
24	L24	Priority Queues	BB	Tx2/pp 193-194,Rx3/pp 161-165
25	L25	Binary Heap – Applications -Heaps	BB	Tx2/pp 195-208,Rx3/pp 296-306 Rx4/pp 396-421
26	L26			
27	L27	Hashing - Hash tables without Linked Lists. Extendible Hashing	BB	Tx2/pp 165-184,Rx4/pp 42-59 Tx2/pp 184-187
UNIT – IV NONLINEAR DATA STRUCTURES				
28	L28	Trees - Binary trees.	BB	Tx2/pp 105-116,Rx3/pp 259-268
29	L29	Tree Traversals.	BB / LCD	Tx2/pp 107-109,Rx3/pp 268-275
30	L30	Binary Search Tree.	BB	Tx2/pp 116-126,Rx3/pp 280-292
31	L31	AVL Trees.	BB	Tx2/pp 126-138,Rx3/pp 408-416
32	L32			
33	L33	Graph Algorithms - Topological sort. Graph Traversals	BB	Tx2/pp 302-306,Rx3/pp 332-340, Rx3/pp 351-363
34	L34	Shortest path Algorithm- Dijkstras Algorithm.	BB	Tx2/pp 306-324,Rx4/pp 580-584
35	L35	Minimum Spanning Tree - Prims Algorithms.	BB	Tx2/pp 329-332,Rx4/pp 575-580
36	L36	Kruskal's Algorithms.	LCD	Tx2/pp 332-335,Rx4/pp 575-580
UNIT – V SORTING AND SEARCHING				
37	L37	Sorting –Bubble Sort.	LCD	Rx3/pp 101-104
38	L38	Insertion Sort.	LCD	Tx2/pp 236-238,Rx3/pp 105-106
39	L39	Shell sort.	BB	Tx2/pp 238-242,Rx3/pp 115-116
40	L40	Heap sort.	BB	Tx2/pp 242-246
41	L41	Merge sort.	BB / LCD	Tx2/pp 246-251,Rx3/pp 107-112
42	L42	Quick sort.	BB	Tx2/pp 251-263,Rx3/pp 112-115
43	L43	Bucket sort.	BB	Tx2/pp 70-72
44	L44	Searching Techniques – Linear Search.	LCD	Rx3/pp 86-92,Rx4/pp 40-42
45	L45	Binary Search.	LCD	

UNIT – I – INTRODUCTION TO C++
PART – A

1.What is an object oriented language?(Remembering) (Co1)

Object oriented language is a programming language that supports Object Oriented Programming. Object Oriented Programming languages should- Support classes/objects, Support inheritance/polymorphism, Support information hiding/encapsulation.

2. List the benefits of OOPS.(Remembering) (Co1)(June 2017)

Through inheritance, we can eliminate redundant code and extend the use of existing classes. The principle of data hiding helps the programmer to build secure programs. It is possible to have multiple instances of an object to co-exist without any interference. Object oriented systems can be easily upgraded from small to large systems. Software complexity can be easily managed.

3. Summarize the applications of OOPS.(Understanding) (Co1)

Real time systems ,Simulation and modeling, Object oriented data bases AI and expert systems, Neural networks and Parallel programming, Decision support and Office automation systems, CAD/CAM systems

4.outline the structure of a C+ + program. (Dec 2009/Dec 2010)(Understanding) (Co1)

Include files
Class declaration
Member function definition
Main function program

5. What are the different types of binding of object with pointer?(Remembering) (Co1)

When you have a pointer to an object, the object may actually be of a class that is derived from the class of the pointer. Thus there are two types: 1.Static binding 2. Dynamic binding

6. Define static binding? (Remembering) (Co1)

Static typing means that the legality of a member function invocation is checked at the earliest possible moment: by the compiler at compile time. The compiler uses the static type of the pointer to determine whether the member function invocation is legal.

7. What is Dynamic binding?(Remembering) (Co1)

Dynamic binding means that the address of the code in a member function invocation is determined at the last possible moment: based on the dynamic type of the object at run time. It is called "dynamic binding" because the binding to the code that actually gets called is accomplished dynamically (at run time). Dynamic binding is a result of virtual functions.

8. Define Class. (Remembering) (Co1)

A Class is a collection of objects of similar type. The classes are user-defined data types and behave like built-in types of a programming language. A class is a way to bind the data and its associated functions together. A class is a user-defined data type with a template that serves to define its properties. A class is a blueprint that defines the variables & the methods common to all objects of a certain kind.

9. What do you mean by object? (Dec 2010)(Remembering) (Co1)

Objects are the basic runtime entities. An object is a software bundle of variables and related methods. An object is characterized by its identity, its state & its behavior. The state of an object is given by its variables. The identity of an object distinguishes objects. Behavior of an object is specified by its methods, objects can reflect real world components that are part of the problem.

10. Compare classes & objects? (Dec2010)(understanding) Co1)Objects Vs Classes

- * Objects are instances of a class
- *Classes are implemented by the programmer
- *Objects are created during the execution of a program.
- *All objects have the same behavior (ie methods); methods are specified for an entire class.
- *All objects of a class have the same variables("members"), but with different values

11. What is meant by Encapsulation?(Remembering) (Co1) (june 2015/Dec 2017)

The wrapping up of data and function into a single unit (class) is known as Encapsulation.

12. How does the memory space occupied by the different objects of a class? (dec 2013)(Remembering) (Co1)

1. The member functions are created and placed in the memory space only once when they are defined as a part of a class specification. 2. All the object belonging to that class use the same member function, no separate space is allocated for member functions when the objects are created.

13. Identify the general form of a class definition in C++ . or how to specify a class?(Applying) (co1) (jan 2016)

```

Class class_name
{
    Private:
        Variable declaration;
        Function declaration;
    Public:
        Variable declarations;
        Function declarations;
};

```

14. Define Data abstraction? (Remembering) (Co1)

Abstraction refers to the act of representation of essential features without including the background details or explanations. Classes use the concept of abstraction & are defined as a list of abstraction attributes such as size, weight & cost & functions to operate on these attributes.

15. What do you mean by message passing?(Remembering) (Co1)

Objects communicate with one another by sending and receiving information.

A message for an object is a request for execution of a procedure, and therefore will invoke a function in the receiving object that generates the desired result.

Message passing involves specifying the name of the object, the name of the function and the information to be sent.

16. Why do we need the preprocessor directive #include <iostream.h>?(Remembering) (Co1)

This directive causes the preprocessor to add the contents of the iostream.h file to the program. It contains the declarations for the identifier cout and the operator <<. It contains function prototypes for the standard input output functions.

17. make use of return statement in main() function?(Understanding) (Co1)

In C++, main() returns an integer type value to the operating system. Therefore, every main() in C++ should end with a return(0) statement; otherwise a warning or an error might occur.

18. How does a main() function in C++ differ from main() in C?(Remembering) (Co1)

In C++, main() returns an integer type value to the operating system but in C , Main() returns nothing to operating system by default.

19. Justify unsigned int can be twice as large as the signed int. (Applying) (Co1)

In Signed int , Most significant bit(MSB) is reserved for representing the sign, but in unsigned int no bit is reserved for storing the sign. So using unsigned int we can represent the values twice as large as the signed int. That is range of values represented by:

Signed int : -32768 to 32767 & Unsigned int : 0 to 65535

20. What is the use of modifiers used in C++?(Remembering) (Co1)

Modifiers are used to alter the meaning of the base type to fit various situations more precisely. The list of modifiers is: i. signed, ii. Unsigned, iii. Long, iv. short

21. What is the use of exit() function?(Remembering) (Co1)

The exit() function causes immediate termination of the entire program, forcing a return to the operating system. The general form : Void exit(int return_code);

The value of return code is returned to the calling process, which is usually the operating system. Zero is generally used as a return code to indicate normal program termination.

22. Define function?(Remembering) (Co1)

Functions are the building blocks of C++ and the place where all program activity occurs. The general form is,

```

ret-type function-name(parameter list)
{
    body of the function
}

```

The ret-type specifies the type of data that the function returns. The parameter list is a comma-separated list of variable names and their associated types that receive the values of the

arguments when the function is called. When the function is called the control is transferred to the first statement in the body.

The other statements in, the function body are then executed and control returns to the calling program when the closing brace is encountered

23. What are the two ways to pass arguments to the function?(Remembering) (Co1)

Call by value –

This method copies the value of an argument into the formal parameter of the function.

Call by reference –

This method copies the address of an argument into the formal parameter of the function.

24. Find the value of x after the execution $x=(y=50)+10$; (jan 2014)(Applying) (Co1)

X=60

25. Define the keyword 'delete' (Dec2010)(Remembering) (Co1)

When a data object is no longer needed it should be destroyed to release the memory space for reuse, the general form is, Delete pointer variable;

The pointer variable is the pointer that points a data object created with new. e.g. delete p;

26. classify different memory management operators used in C++?(Understanding) (Co1)

The memory management operators are new and delete. The new and delete operators are used to allocate and deallocate a memory respectively. General form of new : Pointer-variable = new datatype; - General form of delete: delete pointer-variable;

27. What are the different of operators used in C++?(Remembering) (Co1)

Arithmetic operators (+ , - , * , / , %), Relational operators(< , <= , > , >= , ==, !=), Logical operators (&& , || , !), Bitwise operators (& , | , ^ , ~)

28. Compare Structure and Classes(Understanding) (Co1)

Structures	Classes
1. By default the members of the structure are public.	1.By default the members of Class are private.
2. Data hiding is not possible	2.Data hiding is possible
3.structure data type cannot be treated like built-in-type	3.Objects can be treated like built-in-types by means of operator overloading.
4.Structure are used only for holding data	4. Classes are used to hold data and functions
5.Keyword 'struct'	5.Keyword 'class'

29. How objects are created?(Remembering) (Co1)

Once a class has been declared, we can create variables of that type by using the class name .The class variables are known as objects .The necessary memory space is allocated to an object at the time of declaration. Ex : class student // class declaration

```
{
    char name[10];
    int rollno;
public :
    void getdata( );
    void putdata ( );
};
```

student s1, s2; /memory for s1,s2 are created.s1,s2 are objects.

30. How the members of a class can be accessed? (Remembering) (Co1)

The private data of a class can be accessed only through the member functions of that class.The format for calling a member function:**Objectname.function_name(actual _arguments);**

31. What are the two ways of defining member functions?(Remembering) (Co1)

Member functions can be defined in two places:

- i) Outside the class definition ii) Inside the class definition

32. What do you mean by nesting of member functions?(Remembering) (Co1)

A member function can be called by using its name inside another member function of the same class. This is known as nesting of member functions.

33. Explain about static member functions,(understanding)(C02)

A member function can have access to only other static members declared in the same class, A static member function can be called using the class name as follows **Classname :: function name;**

34. What are the ways of passing objects as function arguments?(Remembering) (Co1)

Pass by value : A copy of the entire object is passed to the function, Any changes made to the object inside the function don't affect the objects used to call the function.

Pass by Reference: Only the address of the object is transferred to the function, When an address of the object is passed, the called function works directly on the actual object used in the call. This means that any changes made to the object inside the function will reflect in the actual object.

35. What do you mean by friend function?(Remembering) (Co1)(June 2017)

By declaring a non member function as friend of a class , we can give full rights to access its private data members (i.e.)A friend function although not a member function have full access rights to the private members of the class.

36. List special characteristics of a friend function?(Applying) (C02)

It is not in the scope of the class to which it has been declared as friend, Since it is not in the scope of the class, it cannot be called using the object of the class, It can be invoked like a normal function without the help of any object, Unlike member functions it cannot access the member names directly and has to use an object name and dot membership operator with each member name, Usually it has the object as arguments.

37. What is a const member function?(Remembering) (Co1)

If a member function does not alter any data in the class, then we may declare it as a const member function.

e.g. : void getbalance () const; void mul(int,int) const;

38. What is an abstract class?(dec 2012) (dec 2009)(Remembering) (Co1)

An abstract class is one that is not used to create objects. An abstract class is designed only to act as a base class.

39. In which situations inline expansion may not work? (May 10)p-83(Dec 11)(Applying)(Co1)

- i. For functions returning values, if a loop, a switch, or a goto exists.
- ii. For functions not returning values, if a return statement exists.
- iii.If function contain static variables.
- iv.If inline functions are recursive.

40. What is an Identifier?(Dec 2009)(Remembering) (Co1)

Identifiers refer to the names of variables, functions, arrays, classes etc., created by the programmer.

41.Mention some of the stream classes? (Dec 2010)(applying)(co1)

ios-general input and output,istream-input ,ostream-output ,iostream-input/output stream,streambuf

42.Highlight the advantage of static data members and static function in c++?(May 2012)(dec 2009, 2011) (Analyzing) (Co1) (june 2015)

Static Data member:

- 1.Initialize to zero when the first object of its class is created
- 2.only one copy of member is created for the entire class.
- 3.visible only within the class.

Static Functions:

Can access to only other static members in same class.

Static function can be called using class name.

43. Is it possible to use multiple private, public and protected labeled sections in a class? Specify the default access modifier for classes and members? (dec 2014) (Applying) (Co1)

No. default access modifier of the class is Private.

44. How are the member function defined in a class? (Dec 2015)

Two ways to define a member function i) inside a class ii) outside of a class

PART – B

1. Explain some Tools or strategies used in object oriented design. (Or) Explain the Basics concepts of OOPs and mention merits and demerits of OOps. (june 2015) (jan 2016) (dec 2014) (June 2017) (Dec 2017) (understanding)
2. Mention the special characteristics of Friend function in c++ (or) need of friend function and syntax with example . (pg.no Rx6-115) (Remembering) (Co1) (Dec 2013) (Jan 2016) (Dec 2017)
3. Explain friend function with example. (or) Define friend function . Demonstrate the way to use friend function with an example.. (june 2014) (jan 2015) (Creating) (Co1)
4. create a class called “student” to store marks of the student of a subject in a class. Display the details (ID, name, mark) of the students those who obtained maximum and minimum mark in that subject. (dec 2014) (Creating) (Co1)
5. Mention the concept of member functions and classes with suitable example. (june 2015) (understanding)
6. Define function overloading. Write a program to add 2 integers , 3 integers and (1 int and 1 double) using function overloading. (Dec 2015)
7. Explain with example different types of control structures used in c++ with example? (Dec 2010) (pg.no Rx6-60 to 64) (Understanding) (Co1)
8. Explain the concept of passing array of objects as an argument with an example? (may 2012) (Dec 2017) (pg.no (pg.no Rx6-110 to 114) (Understanding) (Co1)
9. Write a C++ program to perform swapping using private values of two classes. (May/June 2010) (jun 2009) (pg.no Rx6-118) (Applying) (Co1)
10. What is an inline function? Explain the situations where inline expansion may not work? (may 2010) (pg.no Rx6-75 to 77) (Remembering) (Co1)
11. What are friend classes? What are the advantages And limitations of them? Give illustrations to Support your answer. (April 2010) (pg.no Rx6-115 to 118) (Remembering) (Co1)
12. Write a c++ program for employee details using class and objects. (dec 2013) (pg.no Rx6-111) (Applying) (Co1)

UNIT – II – CONSTRUCTORS AND INHERITANCE

PART – A

1. What do you mean by constructor? (dec 2012) (Remembering) (CO2)

Constructor is a member function, It initializes the objects of its class, It has the same name as the class, It is automatically invoked when the objects are created, It constructs the value of the data members of the class, It has no return type.

2. What do you mean by automatic initialization of objects? (Remembering) (CO2)

C++ provides a special member function called the constructor which enables an object to initialize itself when it is created. This is known as automatic initialization.

3. Define default constructor?(Dec 2011) (Remembering) (CO2)

A constructor that accepts no parameters is called default constructor. Ex : The default constructor for Class A is A:: A().

4. What do you mean by Dynamic constructors? (Remembering) (CO2)

The constructors can be used to allocate memory while creating objects. This will enable the system to allocate the right amount of memory for each object, thus resulting in saving of memory. Allocation of memory to objects at the time of their construction is known as dynamic construction of objects. The memory is allocated with the help of 'new' operator.

5. What do you mean by destructor?(jun 2009) (Remembering) (CO2)

Destructor is used to destroy the object that have been created by a constructor, It is a member function, whose name is same as the class name preceded by a tilde, The destructor never takes any arguments nor does it return any value, It will be invoked implicitly by the compiler upon exit from the program to clean up storage. e.g., `~integer () { }`

6. Define Operator Overloading.(Dec 2012,2010)(dec 2009) (Remembering) (CO2)

An operator overloading may exhibit different behaviours in different instances. The behaviour depends on the types of data used in the operation. For example consider the 'addition' for two numbers the operation will generate a sum. If the operands are string then the operation would produce a third string by concatenation.

The process of making an operator to exhibit different behaviours in different instances is known as operator overloading.

7. Identify type of constructor gets called when create an array of objects? (Applying) (CO2)

Default constructor

8. What operators can/cannot be overloaded?(jun 2009)(dec 2012)(jun 2012) (june 2015) (Remembering) (CO2)

Most can be overloaded. The only C operators that can't be are . and ?: (and size of, which is technically an operator). C++ adds a few of its own operators, most of which can be overloaded except :: and .*.

9. Define virtual base class?(Remembering) (CO2)

Multipath inheritance may lead to duplication of inherited members from a 'grandparent' base class. This may be avoided by making the common base class as virtual base class.

When a class is made a virtual base class, only one copy of that class is inherited, regardless of how many inheritance paths exist between the virtual base class and a derived class.

10. What is function overloading? .(Dec 2009) (june 2014)(Remembering) (CO2)

A single function name can be used to handle different number and different types of arguments. Using a single function name to perform different types of tasks is known as function overloading.

11. What is dot operator and what for it is used? (Dec 2010)(Remembering) (CO2)

An object is passed by reference; the member access operator remains the dot operator. Dot operators are used to access the member variables, Access the public members of a class.

12. what is the value of null pointer in c++?(dec 2014)(Remembering) (CO2)Zero.**13. Define virtual function.(jun 2012)/(dec/jan 2013/2014)(dec 2010)(Remembering) (CO2)**

When we use the same function name both in the base and derived classes ,also the function in base class is declared as virtual ,C++ determines what function to use at runtime based on the object pointed to by the base pointer rather than the type of pointer. Thus making the base pointer to point to different objects we can execute different versions of the virtual function.

14. What are pointers?(Dec 2012)(Remembering) (CO2)

A pointer is a variable that holds a memory address. This address is the location of another object in memory.

The general form of declaring a pointer variable is : Type *name ;

Where type is the base type of the pointer and may be any valid type. The name of the pointer variable is specified by name. The base type of the pointer defines what type of variables the pointer can point to.

Example: `int * p; //p is a pointer to int values.`

15. What is meant by manipulators?(Understanding)(C04)

Manipulators are operators that are used to format the data display. The most commonly used manipulators are endl and setw. The endl manipulator is used to insert newline character.

Eg. `cout<<"welcome"<<endl;`

The setw manipulator is used to specify a common field width for all numbers and force them to be printed right justified.

Eg. `cout<<setw(5)<<sum;`

16. What is meant by inheritance?(Remembering) (CO2)

Inheritance is a mechanism by which one class can inherit the properties of another. Inheritance allows a hierarchy of classes to be built, moving from the most general to the most specific. When one class is inherited by another, the class that is inherited is called the base class. The inheriting class is called the derived class

17. Analyze how virtual and non-virtual member functions are called? (Analzing)(CO2)

Non-virtual member functions are resolved statically. That is, the member function is selected statically (at compile-time) based on the type of the pointer (or reference) to the object. In contrast, virtual member functions are resolved dynamically (at run-time). That is, the member function is selected dynamically (at run-time) based on the type of the object, not the type of the pointer/reference to that object. This is called "dynamic binding."n 2016)

18. What is pure virtual function?(Remembering) (CO2) (Jan 2016) (june 2015/June 2017)

A pure virtual function has no definition relative to the base class. Only the function's prototype is included. To make a pure virtual function, use this general form: `virtual type funcname(parameter-list)=0;`

The key part of this declaration is the setting of the function equal to Zero .When a virtual function is made pure, it forces any derived class to override it.

19. Classify inheritance? (Analyzing)(CO2)

Various types of inheritance: Single inheritance -Derived class with only one base class is called single inheritance, Multiple inheritance -derived class with several base classes is called multiple inheritance, Hierarchical inheritance: The traits of one class may be inherited by more than one class, Multilevel inheritance: The mechanism of deriving a class from another derived class is known as multilevel inheritance.,Hybrid Inheritance.

20. Define Hybrid inheritance? (Remembering) (CO2)Hybrid inheritance is a combination of two or more types of inheritance. e.g., combination of multilevel and multiple inheritance.

21. Define pure virtual function?(Remembering) (CO2)

A pure virtual function is a function declared in a base class that has no definition relative to the base class. In such cases, the compiler requires each derived class to either define the function or re declare it as a pure virtual function - A virtual function, equated to zero is called a pure virtual function, A class containing such pure function is called as "abstract class".

22. What are pointer operators?(Remembering) (CO2)

The pointer operators are * and &.

The & is a unary operator that returns the memory address of its operand.

The * is a unary operator that returns the value located at the address that follows.

Example:

```
char*p; // declaration of pointer p
char c='a';
p=&c; //address of variable c is assigned to pointer p
cout<<"p="<<*p; // output: *p=a
```

23. Define polymorphism. List its types. (june 2014)(Understanding) (CO2)

Polymorphism means the ability to take more than one form. For example, an operation may exhibit different behavior in different instances. The behavior depends upon the types of data used in the operation. There are two types of polymorphism

Compile time polymorphism,

Run time polymorphism

24. Define Compile time polymorphism / Early Binding / static Binding.(Remembering) (CO2)

Compile time polymorphism is implemented by using the overloaded functions and overloaded operators.

The overloaded operators or functions are selected for invocation by matching arguments both by type and number. This information is known to the compiler at the compile time therefore the compiler is able to select the appropriate function for a particular function call at the compile time itself. This is called as 'Early Binding' or 'Static Binding' or 'Static Linking' or 'Compile time polymorphism'. Early binding simply means an object is bound to its function call at the compile time.

25. Define Runtime Polymorphism.(Remembering) (CO2)

At runtime, when it is known what class objects are under consideration, the appropriate version of the function is invoked. Since the function is linked with a particular class much later after its compilation, this process is termed as 'late binding'. It is also known as dynamic binding because the selection of the appropriate function is done dynamically at run time. This runtime polymorphism can be achieved by the use of pointers to objects and virtual functions.

26. What do you mean by reusability?(Remembering) (CO2)

The process of adding additional features to an existing class without modifying it is known as 'Reusability'. The reusability is achieved through inheritance. This is possible by deriving a new class from an existing class. The new class will have the combined features of both the classes.

27.What is virtual function?Why it is used? .(Dec 2009)(dec 2013)(Applying) (CO2)

If the appropriate member function could be selected while the program is running. This is known as run time polymorphism. C++ supports a mechanism known as virtual function to achieve run time polymorphism.

28.What is put() and get() functions? .(Dec 2009)(Understanding) (CO2)

The classes `istream` and `ostream` define two member functions `get()` and `put()` respectively to handle the single character input/output operations.

29. What do you mean by this pointer in C++? Give an example. (Dec 2009)(Remembering) (CO2)

C++ uses a unique keyword called `this` to represent an object that invokes a member function.

```
class ABC
{
    int a;
    .....
};
```

The private variable 'a' can be used directly inside a member function like `a = 123;` or `this->a = 123;`

30.How exception is handled by c++.(May 10)(jun 2009)(Remembering) (CO2)

- i. The keyword **try** is used to preface a block of statements which may generate exceptions.
- ii. When an exception is detected, it is thrown using a **throw** statement in the try block.
- iii. A catch block defined by the keyword **catch** 'catches' the exception 'thrown' by the throw statement in the try block and handles it appropriately.

31. List out the different type of inheritance. (Dec 2010)(Understanding) (CO2)

Single inheritance, multiple inheritance, multi level inheritance, hierarchical and hybrid inheritance.

32. What are the insertion and extraction operator, give their function. (Dec 2010)(Understanding) (CO2)

The operator `<<` (output operator) is called insertion or put to operator, it inserts (or sends) the contents of the variable.

The operator >> (input operator) is called the extraction operator, it extract (or get from) the value from the key board and assigns it to the variable.

33. What is exception? What information an exception may contain? (Dec 2010)(Remembering) (CO2)

Exceptions are runtime time anomalies or unusual conditions that a program may encounter while executing. Anomalies might include conditions such as division by 0, access to an array outside of its bounds or running out of memory of disk space.

34. What are the types of exceptions?(Remembering) (CO2)Synchronous & Asynchronous exceptions.

35. Name the three keywords used in exception handling.(Remembering) (CO2) Try, Throw, Catch.

36. what is the scope resolution operator (::) and how can it be used for global variable?(may2010)(Applying) (Co1)This operator allows access to the global version of the variable. It can be used to uncover the hidden variable:: variable name

37. what is multiple inheritance?(dec 2012)(Remembering) (CO2) class can be derived from several base class.

38.New operator – Explain (Dec2010)(Remembering) (CO2)

An object can be created by using the keyword new and destroyed by using the keyword delete as and when required.

The general form: Pointer variable = new data type;

Here pointer variable is a pointer of type data type. The new operator allocates sufficient memory to hold a data object of type data and returns the address of the object. The pointer variable holds the address of memory space allocated. **E.g : int *p = new int;**

39. Define the keyword ‘protected’?(Remembering) (Co1)

A member declared as protected is accessible by the member function within its class and any class immediately derived from it. It cannot be accessed by the functions outside these two classes.

40. Give the visibility of base class members in derived class?(Understanding) (Co1)

Base Class Visibility	Derived Class Visibility		
	Public derivation	Private derivation	Protected derivation
Private	Not inherited	Not inherited	Not inherited
Protected	Protected	Private	Protected
Public	Public	Private	Protected

41.Name the three type of situations which may arise in the data conversion between incompatible types? (dec 2013)(Analyzing)(CO2)Conversion from base to class type, Class to base type, Class to another class type

42.How many constructor and destructor will be called in the following program?(Remembering) (CO2)(dec2014)

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

```
Class a
```

```
{
```

```
Public: a() {cout<<"cnst called";}
```

```
~a() {cout<<"dest called";}
```

```
int main()
```

```
{
```

```
A* Objectarray=new A[3];
```

```
Delete [] Objectarray; return 0;}
```

```
} 3 times constructor. One time destructor
```

- 43. List the types of constructor. (Dec 2015)** 1. Default constructor 2. Parameterized constructor 3. Dynamic constructor 4. Copy constructor

PART – B

1. Overload the Operator “+” in C++ to add two complex numbers. (Or) Write a program to add 2 complex numbers using operator overloading. What are the operations that cannot be overloaded in C++? **(Applying) (co2) (Dec2013) (june2015)(June 2017)**
2. What is exception handling? Explain the mechanism of it. (Or) Elaborate words the use of try, throw, catch in C++. **(Dec2013, Dec 2015)(jun 2012)(Dec 2017)(pg.no Rx6-344 to 356 (Remembering) (co2))**
3. What is inheritance? Mention its types. Write a C++ program to illustrate the concept of multiple inheritances? **(May 2012) (dec 2013)(pg.no Rx6- 181 to 205)(Remembering) (CO2)**
4. Write a c++ Program using class to implement function overloading. **(june 2014) (Dec 2010)(Dec 2011 (co2))**
5. Explain with an example program the usage of Virtual function. **(Dec 2009) (jan 2015) (June 2017)(pg.no Rx6- 246 to 247)(Remembering) (CO2)**
6. Overload “-” Operator in C++ to find distance between the two points in a quadrant. **(C) (Jan 2015)(CO2)**
7. Illustrate the working of constructor and destructor with an example. (pg.no Rx6- 131 to 146) **(june 2015)(Dec 2017)(co2)**
8. How do you construct and destruct an object? Discuss the various categories of constructors and destructors with example. **(dec 2010) (may 2012)(pg.no Rx6- 132 to 150)(Remembering) (CO2)**
9. Explain the various features of operator overloading and unary/binary operators with example. **(Dec 2009)(Understanding) (CO2)(pg.no Rx6- 157 to 161)**
10. What is polymorphism and how it can be implemented in C++ and virtual functions. **(April 2010) (Nov/Dec 2011)(pg.no Rx6-248 to 249)(Understanding) (CO2)**
11. What is operator overloading? Overload the numerical operators ‘+’ and ‘/’ for complex numbers. **(may 2010)(pg.no Rx6-156 to 160)(Remembering) (CO2)**
12. List out the rules for overloading operators. Write a program to illustrate 2overloading **(pg.no Rx6-170)(Understanding) (CO2)**

UNIT-III – DATA STRUCTURES & ALGORITHMS

PART-A

- 1. Define data structures?(Jan 2016)(Remembering) (Co3)**

A data structure is a mathematical or logical way of organizing data in the memory that consider not only the items stored but also the relationship to each other and also it is characterized by accessing functions.

- 2. What is meant by problem solving?(Remembering) (Co3)**

Problem solving is a creative process, which needs systemization and mechanization.

- 3. List examples for ?(Understanding) (Co3)**

Stacks, Queue, Linked list, Trees, graphs

- 4. Define Algorithm?(june 2014)(Remembering) (Co3)**

Algorithm is a solution to a problem independent of programming language. It consist of set of finite steps which, when carried out for a given set of inputs, produce the corresponding output and terminate in a finite time.

5. Define Program?(Remembering) (Co3)

Set of instructions to find the solution to a problem. It is expressed in a programming language in an explicit and unambiguous manner.

6. What are the features of an efficient algorithm?(Remembering) (Co3)(June 2017)

- Free of ambiguity, Efficient in execution time, Concise and compact, Completeness
- Definiteness, Finiteness

7. List any four applications of data structures?(Understanding) (Co3)

- Compiler design, Operating System, Database Management system, Network analysis

8. Mention the types of bugs that may arise in a program?(Remembering) (Co3)

The different types of bugs that can arise in a program are

- Syntactic error, Semantic error, Logical error

9. What is meant by an abstract data type (ADT)? List some of the ADTs. (Dec 2010)(Remembering) (Co3)

An ADT is a set of operation. Abstract data types are mathematical abstractions. Objects such as list, set and graph along their operations can be viewed as ADT's.

10. What is meant by list ADT?(Remembering) (Co3)

List ADT is a sequential storage structure. General list of the form $a_1, a_2, a_3, \dots, a_n$ and the size of the list is 'n'. Any element in the list at the position i is defined to be a_i , a_{i+1} the successor of a_i and a_{i-1} is the predecessor of a_i .

11. What are the various operations done under list ADT?(Remembering) (Co3)

- Print list • Insert • Make empty • Remove • Next • Previous • Find kth

12. What are the various ways of implementation of list ADT? (May / June 2010)(Remembering) (Co3)

- Simple array implementation of list • Linked list implementation of list

13. What are the advantages in the array implementation of list?(Understanding) (Co3)

- a) Print list operation can be carried out at the linear time,
- b) Find Kth operation takes a constant time

14. What is a linked list?(Remembering) (Co3)

Linked list is a kind of series of data structures, which are not necessarily adjacent in memory. Each structure contains the element and a pointer to a record containing its successor.

15. What is a doubly linked list?(Jan 2016)(Remembering) (Co3)

In a simple linked list, there will be one pointer named as 'NEXT POINTER' to point the next element, where as in a doubly linked list, there will be two pointers one to point the next element and the other to point the previous element location.

16. Show the content of the stack after the execution of each of the following operation, PUSH (A), PUSH (B), POP, POP, PUSH (C) ,POP (Dec 2013)(Applying) (Co3)

Stack is empty

17. Give some examples for linear data structures?(Applying) (Co3)

- Stack
- Queue

18. What is a stack?(Remembering) (Co3)

Stack is a data structure in which both insertion and deletion occur at one end only. Stack is maintained with a single pointer to the top of the list of elements. The other name of stack is Last-in -First-out list.

19. Write postfix from of the expression $-A+B-C+D$?(Remembering) (Co3)

$A-B+C-D+$

20. How do you test for an empty queue?

To test for an empty queue, we have to check whether $READ=HEAD$ where REAR is a pointer pointing to the last node in a queue and HEAD is a pointer that pointer to the dummy header. In the case of array implementation of queue, the condition to be checked for an empty queue is $READ < FRONT$.

21. What are the postfix and prefix forms of the expression?(Applying) (Co3)(Dec 2017)

$A+B*(C-D)/(P-R)$, Postfix form: $ABCD-*PR-/+$, Prefix form: $+A/*B-CD-PR$

22. Explain the usage of stack in recursive algorithm implementation?(Remembering) (Co3)

In recursive algorithms, stack data structures is used to store the return address when a recursive call is encountered and also to store the values of all the parameters essential to the current state of the procedure.

23. Write down the operations that can be done with queue data structure?(Remembering) (Co3)

Queue is a first - in -first out list. The operations that can be done with queue are addition and deletion.

24. What is a circular queue?(Remembering) (Co3)

The queue, which wraps around upon reaching the end of the array is called as circular queue.

25. What is the need for hashing? Define hash function?(Remembering) (Co3)

Hashing is used to perform insertions, deletions and find in constant average time.

Hash function takes an identifier and computes the address of that identifier in the hash table using some function.

26. List out the different types of hashing functions?(Understanding) (Co3)

- a. The division method, b. The mind square method, c. The folding method,
- d. Multiplicative hashing, e. Digit analysis

27. What are the problems in hashing? (Remembering) (Co3)

a. Collision, b. Overflow
A heap in which the parent has a larger key than the child's is called a max heap.

A heap in which the parent has a smaller key than the child's is called a min heap.

30. Write an algorithm for simple hash function. (May / June 2010)(Applying) (Co3)

```
typedef unsigned int Index;
Index
Hash( const char *Key, int TableSize )
{ unsigned int HashVal = 0;
  While( *Key != '\0' )
    HashVal += *Key++;
  Return HashVal % TableSize; }
```

31. What is an array? Give a syntax of it. (Dec 2009)(Remembering) (Co3)

Group of elements in a memory is called an array. In C++, the size should be one larger than the number of characters in the string.

char string[4]="xyz"; o.k for C++

32. What is Heap? (Dec 2009)(dec 2012)(dec 2010)(june 2015)(Remembering) (Co3)

A particular application seems to require a special kind of queue, known as a priority queue.

Ex.Binary

heap, d-Heaps, Leftst heaps , skew heaps and binomial queues.

33. What is the name of the basic heap operations and its properties? (May 10)(Remembering) (Co3)

The efficient way of implementing priority queue is binary heap. Binary heap is merely referred as heaps, Heaps have two properties namely 1. Structure property. 2. Heap order property.

34. State the advantage of Circular list over Doubly Linked List (Dec2003)(Understanding)(Co3)

A popular conversion is to have the last cell keep a pointer back to the first. This can be done with or without a header and also can be done with doubly linked lists.

35. Compare and contrast stack and queue? (Dec 2009)(Understanding)(Co3)

STACK	QUEUE
Stack is a List	Queues are Lists
Insertions and deletions can be performed in only	Insertion is one end and deletion is another

one position namely ,the end of the list, called the top	end.
Push and Pop are funtemental operations are used to insert and deletes the elementls repectively	Enqueue-Inserts an element at the end of the list(called the rear) Dequeue , which deletes (and returns) the element at the start of the list(front)
LIFO (Last in First out operation)	FIFO

36. Write the prefix and postfix form of the expression (A+B)(C-D) (May 10) (Applying)(Co3)

i. pre AB+CD-*

ii. Post *+AB-CD

37. Convert into postfix notation 4*2+7-3. Ans :12(Applying)(Co3)

38. What is dequeue?(nov dec 2009)(Remembering) (Co3)

Dequeue deletes (and returns) the element at the start of the list(front).(Delete the data in the queue.

39. Write any two data structure used in operating system?(april may 2010)stack, queue

40. What are the representations of Big and small 'O' notations? (april may 2010)(Remembering) (Co3)

Big-oh (O(N)) represents the possibility that the growth rates are the same.

Small-oh (o(N)) represents the possibility that the growth rates are the different.

41. List out the performance measures of an algorithm? (May 2010)(Remembering) (Co3)

Best Case, Worst Case, Avarage Case

42. What are the limitations of linear queue? How are they overcome using circular Queue?(Remembering) (Co3)

The implementation of circular queue is same as that of linear queue, except that the last position is connected to the first position of the queue. so the front and rear end moves circularly.

43. what is meant by underflow and overflow condition in a stack?(june 2015)(Remembering) (Co3)

When the condition stack is empty top=-1 is called underflow conndition. When the stack is full top=N-1, is the total size of stzck is called overflow condtion.

44. Mention the use of stack in various applications?(dec 2014)(Understanding) (Co3)

Towers of Hanoi, Infix to postfix, function calls, balancing symbols

PART-B

1. Explain the linked list and array implementation of stack ADT, write program to implement stack operations?(May,Dec 2010) (may 2012) (dec 2012) (dec 2013) (jan 2015)(pg.no.Tx1-78 to 80) (Understanding) (Co3)
2. Explain priority queue and how are binary heaps used in that? (Or) Explain binary heap operations with examples . (dec2013, june 2014) (Dec 2017)(pg.no.Tx1- 195 to 202) (Understanding) (Co3)
3. (may 2010) (dec 2012)(pg.no.Tx1- 195 to 202)(dec 2013) (Understanding) (Co3)
4. Explain the linked list implementation of list ADT? (june 2014)(june 2015)(pg.no.Tx1- 70 to 77) (Understanding) (Co3)
5. All possible code to implement queue ADT in detail?(jan 2015,june 2015) (Dec 2010)(June/Dec 2017)(pg.no.Tx1- 95 to 100) (Understanding) (Co3)
6. Define Hashing. Name some hashing techniques. & Compare the various hashing Techniques.(jan 2015)(may 2010)(jan 2015)(June 2017)(pg.no.Tx1- 166 to 180)(Remembering) (Co3)

7. What are the various operations that can be performed on stacks and queues ? Explain the algorithm for each. (Dec2009)(Dec 2010). (Nov/Dec 2011). (pg.no.Tx1- 79 to 84) (Evaluating) (Co3)
8. Give a procedure to convert an infix expression $a+b*c+(d*e+f)*g$ to postfix notation.(May/Jun10)(pg.no.Tx1- 78 to 100)(Applying) (Co3)
9. Construct a heap for the key set:42,23,74,11,65,58,94,36,99,87(may 2012) (pg.no.Tx1- 202) (Creating) (Co3)
10. Convert the expression $((A+B)*C-(D-E)^(F+G))$ to equivalent prefix and postfix notations. (may 2010) (pg.no.Tx1-113)(Remembering) (Co3)

UNIT-IV – NONLINEAR DATA STRUCTURES

PART-A

1. Define non-linear data structure.(Remembering) (Co4)

Data structure which is capable of expressing more complex relationship than that of physical adjacency is called non-linear data structure.

2. Define tree&leaf.(Remembering) (Co4)

A tree is a data structure, which represents hierarchical relationship between individual data items.In a directed tree any node which has out degree 0 is called a terminal node or a leaf.

3. Define directed tree?(Remembering) (Co4)

Directed tree is an acyclic diagram which has one node called its root with in degree 0 while all other nodes have in degree 1.

4. Illustrate a ordered tree?(Understanding) (Co4)

In a directed tree if the ordering of the nodes at each level is prescribed then such a tree is called ordered tree.

5. List applications of binary tree?(Remembering) (Co4)(June 2017)

- a. File index schemes
- b. Hierarchical database management system

6. What is meant by traversing and types of traversing?(Understanding) (Co4)

Traversing a tree means processing it in such a way, that each node is visited only once.

- a. Pre-order traversal-yields prefix form of expression.
- b. In-order traversal-yields infix form of expression.
- c. Post-order traversal-yields postfix form of expression.

7. What are the two methods of binary tree implementation?(Understanding) (Co4)

- a. Linear representation.
- b. Linked representation

8. Define pre-order traversal?(Remembering) (Co4)

Pre-order traversal entails the following steps;

- a. Process the root node
- b. Process the left subtree
- c. Process the right subtree

9. Define post-order traversal?(Remembering) (Co4)

Post order traversal entails the following steps;

- a. Process the left subtree
- b. Process the right subtree
- c. Process the root node

10. Define in -order traversal?(Remembering) (Co4)

In-order traversal entails the following steps;

- a. Process the left subtree
- b. Process the root node
- c. Process the right subtree

11. What is an AVL tree? (Dec 2009, Dec 2010,Jan 2016)(Understanding) (Co4)

An AVL (Adelson-Velskii and Landis) tree is a binary search tree with a balance condition. The balance condition is easy to maintain, and it ensures that the depth of the tree is $O(\log N)$.

12. What is a balance factor in AVL trees? (june 2014) (Remembering) (Co4)

Balance factor of a node is defined to be the difference between the height of the node's left subtree and the height of the node's right subtree.

13. Define pivot node?(Remembering) (Co4)

The node to be inserted travel down the appropriate branch track along the way of the deepest level node on the branch that has a balance factor of +1 or -1 is called pivot node.

14. What is the length of the path in a tree?(Remembering) (Co4)

The length of the path is the number of edges on the path. In a tree there is exactly one path from the root to each node.

15. Define expression trees?(Remembering) (Co4)

The leaves of an expression tree are operands such as constants or variable names and the other nodes contain operators.

16. Define graph ?what are the two traversal strategies used in traversing a graph (jan 2016)(Dec 2010)(Remembering) (Co4)

A graph G consist of a nonempty set V which is a set of nodes of the graph, a set E which is the set of edges of the graph, and a mapping from the set for edge E to a set of pairs of elements of V . It can also be represented as $G = (V, E)$. Types: 1.breadth first search 2. Depth first search

17. Define adjacent nodes?(Remembering) (Co4)

Any two nodes which are connected by an edge in a graph are called adjacent nodes. For example, if an edge $x \in E$ is associated with a pair of nodes (u, v) where $u, v \in V$, then we say that the edge x connects the nodes u and v .

18. Identify a directed and undirected graph?(Applying) (Co4)

A graph in which every edge is directed is called a directed graph.

A graph in which every edge is undirected is called a un-directed graph.

19. What is a loop?(Remembering) (Co4)

An edge of a graph which connects to itself is called a loop or sling.

20. What is a simple graph?(Remembering) (Co4)

A simple graph is a graph, which has not more than one edge between a pair of nodes than such a graph is called a simple graph.

21. What is a weighted graph? (Understanding) (Co4)

A graph in which weights are assigned to every edge is called a weighted graph.

22. Define out degree of a graph?(june 2015)(dec 2011)(Remembering) (Co4)

In a directed graph, for any node v , the number of edges which have v as their initial node is called the out degree of the node v .

23. Define indegree of a graph?(Understanding) (Co4)

In a directed graph, for any node v , the number of edges which have v as their terminal node is called the indegree of the node v .

24. Define path in a graph? (Understanding) (Co4) The path in a graph is the route taken to reach terminal node from a starting node.**25. What is a simple path?(Remembering) (Co4)**

A path in a diagram in which the edges are distinct is called a simple path. It is also called as edge simple.

26. What is a cycle or a circuit? (May / June 2010)

A path which originates and ends in the same node is called a cycle or circuit.

27. What is an acyclic graph?(dec 2013)(Remembering) (Co4)

A simple diagram which does not have any cycles is called an acyclic graph.

28. What is meant by strongly connected in a graph?(Understanding) (Co4)

An undirected graph is connected, if there is a path from every vertex to every other vertex. A directed graph with this property is called strongly connected.

29. When is a graph said to be weakly connected?(Understanding) (Co4)

When a directed graph is not strongly connected but the underlying graph is connected, then the graph is said to be weakly connected.

30. Name the different ways of representing a graph?(Understanding) (Co4)

a. Adjacency matrix, b. Adjacency list

31. What is an undirected acyclic graph?(Dec 2013) (Remembering) (Co4)

When every edge in an acyclic graph is undirected, it is called an undirected acyclic graph. It is also called as undirected forest.

32. What is a minimum spanning tree? (june 2014) (Dec 2010)(June 2017)(Applying) (Co4)

A minimum spanning tree of an undirected graph G is a tree formed from graph edges that connects all the vertices of G at the lowest total cost.

33. Define an NP- complete problem? (May/Jun2010)(Remembering) (Co4)

Among all the problems known to be in NP, there is a subset, known as the NP- complete problems, which contains the hardest.

34. Write a definition of Topological sort. (May / June 2010, Jan 2016)(Remembering) (Co4)

A topological sort is an ordering of vertices in a directed acyclic graph, such that if there is a path from v_i to v_j , then v_j appears after v_i in the ordering.

35. Define root.(Dec 2009 & May / June 2010)(Remembering) (Co4)

A tree consists of a distinguished node r , called the root, and zero or more nonempty (sub) trees T_1, T_2, \dots, T_k , each of whose roots are connected by a directed edge from r .

37. Give any two applications of Graphs. . (Dec 2009)(Understanding) (Co4)

- i. Airline route map- undirected graph
- ii. Flowcharts- Directed graph
- iii. A binary relation- Directed graph
- iv. Computer networks
- v. An electrical circuit.

38. Why a minimum spanning is termed so?(Dec 2009)(Evaluating) (Co4)

A Minimum Spanning Tree (MST) of an undirected graph G is a tree formed from graph edges that

connects all vertices of G at lowest cost ($T=V-1$). MST for undirected graph are i) Prim's algorithm

, ii) Kruskal's algorithm app: i. Wiring a house with a minimum of cable. ii. Cheapest cost tour of travelling salesman. iii. Networking the PC's with low cost.

39. What is complete binary tree? (May 10) & (Nov/Dec 2011)(Evaluating) (Co4)

A binary tree is a tree in which no node can have more than two children. All nodes have two Children is called as complete binary tree.

40. When does the graph become tree?(Nov Dec 2009)(Applying) (Co4)

Graph is a tree formed from graph edges that connects all the vertices of G at lowest total cost.

41. What is meant by adjacent matrix?(dec 2010)(Understanding (Co4)

A graph can be represented as adjacency matrix. A graph $G = (V, E)$ with n vertices is an $n \times n$ matrix. It can be represented as two dimensional arrays. ($A_{ij}=1$ if there is an edge, $A_{ij}=0$ if there is an no edge)

42. State the properties of binary search tree?(Applying) (Co4)

1. The key values in the left sub tree are less than the root and key values in the right sub tree is greater than the root. 2. It is direct link with both left and right child 3. It is easy, traverse in only one side.

43. For what type of graphs spanning trees do not exist?(june 2014)(may 2012)(Applying) (Co4)

Cyclic Directed graph

44. How to build a heap tree? (dec 2013)(Understanding) (Co4)

A heap should be complete binary tree which is completely filled binary tree exception with the possible of the bottom level, which is filled from left to right.

45. How to construct an expression tree?(dec 2013)(Creating) (Co4)

Convert the given expression into postfix. Check the symbol is an operand or operator. In this tree left nodes are operands and internal nodes are operators.

46. Mention the use of rotations in AVL tree. (dec 2013) (Understanding) (Co4)

In AVL tree violations occur in 4 conditions. These violations can be overcome by two rotations, single rotation, Double rotations.

47. How many trees are possible with the 3 nodes? (Applying) (Co4)

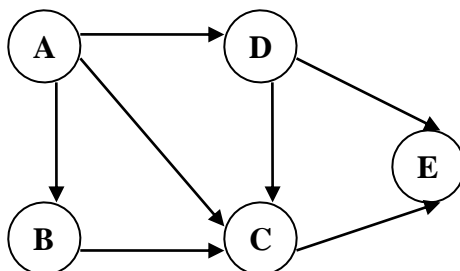
3 trees are possible with 3 nodes. without sub tree.

48. Difference between the binary tree and binary search tree? (Dec 2014) (Understanding) (Co4)

Binary tree	Binary search tree
Binary tree is a complete tree.	Binary search tree right child are greater than root and left sub tree are less than the root.
Searching is difficult.	Searching an element is easy.
Inserting an element is easy	Inserting an element is difficult. It takes time

PART-B

1. What is a binary tree? Also explain the binary tree traversal algorithm with example. (Dec 2009, 2010, 2013, May 2012, Jan 2015, June 2015) (pg.no. Tx1-111 to 113) (Understanding) (Co4)
2. Explain Dijkstra's algorithm for shortest path problem with an example. For any graph. (June 2014) (Dec 2013) (pg.no. Tx1- 311 to 319)
3. Explain the different tree traversals with an application? Write the steps and routine for the Inorder, Preorder and Postorder traversal with example. (June 2014) (Jan 2016) (Dec 2010) (Dec 2013) (pg.no. Tx1-148) (Remembering) (Co4)
4. Define binary search tree? Write the routines to insert and delete a node from BST (May/June 2010, Dec 2010). (Dec 2009) (May 2012) (June 2017) (pg.no. Tx1- 116 to 125) (Understanding) (Co4)
5. Explain Prim's algorithm with an example? (Dec 2010) (Jan 2015) (pg.no. Tx1-330 to 331) (Understanding) (Co4)
6. Write routines for insertion and deletion in a binary tree. (June 2015) (U) (Co4)
7. Discuss Kruskal's and Prim algorithm to find out minimum spanning tree with example. (Dec 2010, 2012, Jan 2016) (June 2017) (pg.no. Tx1- 332) (Remembering) (Co4)
8. Write the pseudocode for Dijkstra's algorithm. (Or) State the principle of Dijkstra's algorithm. (Jun 2009) (Dec 2017). (pg.no. Tx1-311 to 319) (Creating) (Co4)
9. What are the two popular algorithms for computing minimal spanning tree. Explain any one algorithm with neat diagram. (Dec 2009) (Dec 2017). (pg.no. Tx1- 329 to 334) (Remembering) (Co4)
10. Write the pseudocode for unweighted shortest path algorithm using queue. (May/June 2010). (pg.no. Tx1- 307 to 310) (Applying) (Co4)
11. Write the routines for finding the shortest path in a graph. (May/June 2010) & (Nov/Dec 2011). (pg.no. Tx1- 306 to 324) (Creating) (Co4)
12. What is topological sort? Write a routine to perform the topological sort and apply the same to the following graph. (pg.no. Tx1- 309) (Applying) (Co4)

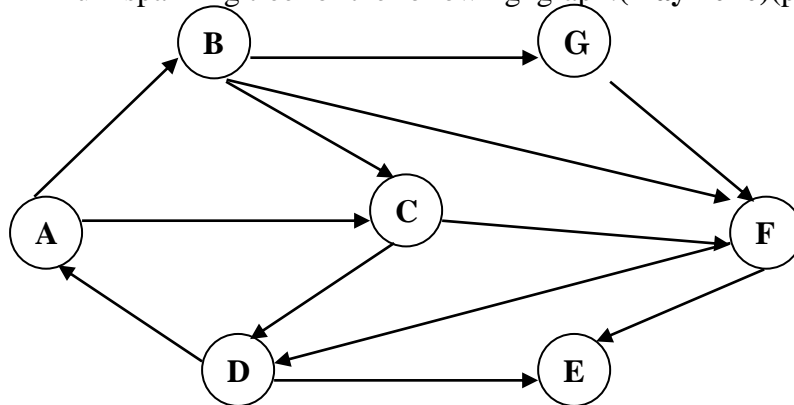


13. Write a shortest path algorithm and apply it to the above figure given. (pg.no.Tx1-306)
(Evaluating) (Co4)

14. Find the following: (Creating) (Co4)

(a) Find the shortest path from A to all other vertices for the following graph.

(b) Find a minimum spanning tree for the following graph. (may 2010) (pg.no.Tx1- 306)



UNIT-V – SORTING & SEARCHING

PART-A

1. What is meant by sorting?(dec 2010,june 2014)(Remembering) (Co5)

Ordering the data in an increasing or decreasing fashion according to some relationship among the data item is called sorting.

2. What are the two categories of sorting? (Dec 2009)(Dec 2015) (Understanding) (Co5)

Internal sort: Data are assumed to be in the high-speed, random access memory.

External sort: Algorithms for sorting large set of data stored on external devices, that is slower devices with restrictions on the way data are accessed is called external sort.

3. List types of external sorting?(Remembering) (Co5)

2-way Merge Sort, K-way balanced Merge Sort, Cascade Merge Sort, Polyphase Merge Sort

4. Define external sorting & internal sorting.(Understanding) (Co5)

External sorting is a process of sorting in which large blocks of data stored in storage devices are moved to the main memory and then sorted.

Internal sorting is a process of sorting the data in the main memory.

5. What are the various factors to be considered in deciding a sorting algorithm?(Remembering) (Co5)

a. Programming time, b. Execution time of the program, c. Memory needed for program environment

6. How to sort elements using Bubble sort? (dec 2013)(Dec 2015)(Understanding) (Co5)

If the element at the lower index is greater than the element at the higher index, the two elements are interchanged. so that the smaller element is placed before the bigger one.

7. What is the main idea behind insertion sort?(Applying) (Co5)

The main idea behind the selection sort is to find the smallest element among in A (I) A (J+1)...A (n) and then interchange it with a (J). This process is then repeated for each value of J.

8. What is the basic of shell sort?(Applying) (Co5)

Instead of sorting the entire array at once, it is first divide the array into smaller segments, which are then separately sorted using the insertion sort.

9. What is the other name for shell sort? -(Remembering) (Co4) Diminishing increment sort.

10. What is the purpose of quick sort? (Understanding) (Co5)

The purpose of the quick sort is to move a data item in the correct direction, just enough for to reach its final place in the array.

11. What is the advantage of quick sort?(Remembering) (Co5)

Quick sort reduces unnecessary swaps and moves an item to a greater distance, in one move.

12. When is a sorting method said to be stable?(Applying) (Co5)

A sorting method is said to be stable, if two data items of matching values are guaranteed to be not rearranged with respect to each other as the algorithm progresses.

13. Name some simple algorithms used in external sorting?(Understanding) (Co5)

a. Multiway merge, b. Polyphase merge, c. Replacement selection

14. What is Merge Sort?(Understanding) (Co5)

Merge sort takes advantage of the ease of merging already sorted lists into a new sorted list. It starts by comparing every two elements (i.e., 1 with 2, then 3 with 4...) and swapping them if the first should come after the second. It then merges each of the resulting lists of two into lists of four, then merges those lists of four, and so on; until at last two lists are merged into the final sorted list.

15. What is an Insertion sort? When can we use it? (Jan 2015 , June 2016, dec 2009/June 2017)(Applying) (Co5)

One of the simplest sorting algorithms is the insertion sort. Insertion sort consists of $N-1$ passes. For pass $P=1$ through $N-1$, insertion sort ensures that the elements in positions through P are in sorted order.

Insertion sort is useful only for small files or very nearly sorted files.

16. What is meant by quick sort? (Dec 2009)(Remembering) (Co5)

Quick sort is the fastest known sorting algorithm. Its average running time is $O(N \log N)$. To partition an array, we choose an element, called a *pivot*, move all smaller elements before the pivot, and move all greater elements after it. This can be done efficiently in linear time and in-place.

17. Write a routine for insertion sort. (May / June 2010) (Remembering) (Co5)

```
void
Insertionsort( ElementType A[ ], int N )
{
    int j, P;
    ElementType Tmp;
    for( P=1; P<N; P++ )
    {
        Tmp = A[P];
        for( j=P; j>0 && A[j-1] > Tmp; j-- )
            A[j] = A[j-1];
        A[j] = Tmp;
    }
}
```

18. What are the ways to pick the pivot element? (May / June 2010)(Applying) (Co5)

1.A wrong way: uniform choice is to use the first element as the pivot.

2.A safe Maneuver : To choose the pivot randomly.

3.Median-of-Three partitioning: The median of a group of N numbers is the $[N/2]$ th largest number. **The best choice of pivot would be the median of the array.**

19. Compare the single source shortest path algorithm with all pairs shortest path algorithm. (Dec 2010)(Understanding) (Co5)

Single source: To find the shortest path from one vertex to all other vertices

All pairs: To find the shortest path between all vertices.

20. Give examples of sorting techniques that is not based on Divide and Conquer algorithm. (Remembering) (Co5)

1. Bubble sort 2. Selection Sort
3. Insertion Sort 4. Radix sort

21. Is merge sort stable algorithm?(Remembering) (Co5)

Yes, merge sort is the stable sorting algorithm. A sorting algorithm is said to be stable if it preserves the ordering of similar (equal) elements after applying sorting method. And merge sort is a method which preserves this kind of ordering. Hence merge sort is a stable sorting algorithm.

22.list classifications of sorting.(Understanding) (Co5)

Sorting: 1. internal sorting 2. External sorting

Internal sorting: 1. Insertion sort (Insertion sort, shell sort) 2. Selection sort(Selection, heap), 3. Exchange Sort (Bubble, Quick).

External sorting (merges): Natural, Balanced and Polyphase.

23. Is quick sort stable algorithm?(Applying) (Co5)

No, quick sort is not a stable sorting algorithm because after applying this sorting method the ordering of similar elements may not be preserved. (since swapping is involved in this sorting with pivot element).

24.What is meant by sequential search?(nov dec 2009 ,June 2015)(Remembering) (Co5)

Sequential search also called as linear search,it is used for searching an array for a particular value.It works by comparing every element of the array one by one in sequential until the match is found.

25. What is meant by bucket sort? Feature of bucket sort?(April/ may 2010)(Dec 2017) (Understanding) (Co5)

Radix sort also called as bucket sort and also most efficient sort among all others. It manages to sort the values without performing any comparison on the input data. The values are ordered on digit position from right to left. this is performed by copying the values into buckets.

PART-B

1. What is sorting? Explain about quick sort algorithm. Sort the following,67,72,78,83,86,62,55,52,46 (Dec 2013,2015) (pg.no.Tx1- 251 to 257)(Remembering) (Co5)
2. Perform bubble sort for the given list of numbers 56,91,35,72,48,68. (dec 2013,June 2015,Jan 2015).(Understanding) (Co5)
3. with example explain the binary search and linear search techniques (Nov/Dec 2012) (dec 2013,June 2015)(June 2017)(Evaluating) (Co5)
4. Explain shell sort with an example? (June 2014)(pg.no.Tx1- 238 to 240)(Understanding) (Co5)
5. Different types of searching techniques with suitable examples.(June 2014,Jan 2015)(co5)
6. Write down the merge sort algorithm and show how merge sort processes the following input. 24, 13, 26, 1, 2,27,38,15. B.38,27,43,3,9,82,10 (May/Jun10,jan 2015,June 2015)(Dec 2017) (pg.no.Tx1- 246)(Remembering) (Co5)
7. Trace the quick sort algorithm for the following list of numbers.90,77,60,99,55,88,66. (nov dec 2011)(June/Dec 2017) (pg.no.Tx1-251)(Applying) (Co5)
8. Show that the resulting running shell sort on the input 9,8,7,6,5,4,3,2,1 using the increments {1,3,5}.(pg.no.Tx1-238)(Understanding) (Co5)
9. Apply quick sort to sort the list E, X, A, M, P, L, E. and Q, U, E, S, T, I, O, N. (pg.no.Tx1- 251)(Understanding) (Co5)
10. Write a merge sort and bubble sort algorithm with suitable example.(dec 2012) (dec 2013) (pg.no.Tx1- 246)(Understanding) (Co5)
11. Perform insertion sort for the given list of numbers 25,37,18,82,55,64,78. (pg.no.Tx1- 236)
12. Sort the following elements using heap sort 25 55 46 35 10 90 84 31. (pg.no.Tx1-242)(Creating) (Co5)

13. Explain the quick sort algorithm. Sort following numbers using quick sort. Show output of each pass. 65 70 75 80 85 60 55 50 45 & 90 43 63 94 55 88 67 83 77 23 (Nov/Dec 2011)(may 2010) (pg.no.Tx1- 251) (Applying) (Co5)

C++ AND DATA STRUCTURES – TUTORIAL QUESTIONS

- Construct an expression tree for the expression $(a + b + c) + ((d * e + 1) * g)$. Give the outputs when you apply preorder, inorder and post order traversals.
- Construct the expression tree for $a + (b * c) - (d * e)$.
- Show that the result of inserting 10, 1, 14, 6, 5, 8, 15, 3, 9, 7, 4, 11, 13 and 2 one at a time, into an initially empty binary heap. Also perform the delete Min operation in the above heap and give the final heap.
- Show that the result of inserting 3, 1, 4, 6, 9, 2, 5, 7 into an empty BST and delete the root node.
- Insert 17, 21, 13, 15, 10, 16, 4, 24, 27, 23, 11, 25, and 26 into an empty BST. Delete 4, 10, 27 and 13 from the tree.
- Construct an empty AVL tree for the elements 2, 1, 4, 5, 3, 9, 6, 7.
- Apply the bubble sort to sort the following: 78, 34, 42, 20, 72.
- Write a procedure for sorting 3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5 using merge sort.
- Explain the quick sort to sort the following numbers 90, 43, 63, 94, 55, 88, 67, 83, 77, 23.
- Write the algorithm for quick sort and use it to sort the following 67, 72, 78, 83, 86, 62, 55, 52, 46.
- Illustrate inserting an element into a heap with the following numbers 10, 7, 21, 3, 5.
- A binary tree has 8 nodes. The in-order and post-order traversals of the tree are given below. Draw the tree and find pre order traversal.
In-Order: F E C H G D B A Post-Order: F C E A B H D G
- Insert an element into a heap sort with the following input 10, 7, 21, 3, 5. Explain the stages of heap sort.
- Search the element 57 from the given sequence using linear search and binary search technique. 1, 2, 3, 9, 11, 13, 17, 25, 57, 90.
- Create a binary search tree (BST) for the following alphabets. Start from an empty binary search tree R, F, G, B, Z, U, P, K, L. Delete keys B, U and L one after the other and show the trees at each stage.
- Explain the Prim's algorithm for finding the MST for Figure 1.

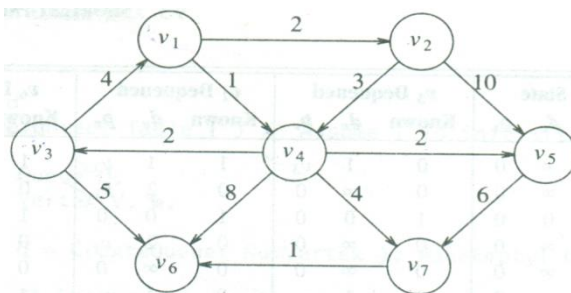


Figure 1

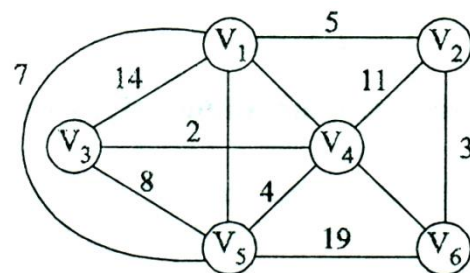
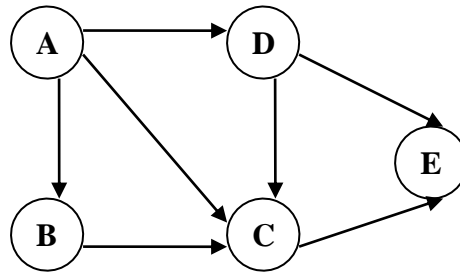


Figure 2

- Apply the Kruskal's algorithm for Figure 1 and find the MST.
- Apply the Prim's algorithm for Figure 2 and find Minimum Spanning Tree.
- Find the minimum spanning tree for the Figure 2 using Kruskal's Algorithm.
- Implement Dijkstras algorithm for Figure 2 and find the shortest path.
- Write the topological sort algorithm and apply the following graph.



23. Create a class 'employee' in which id, age, number of hours worked and wage/hour values will be kept. Write a C++ program to display the salary details of all the employees along with their name and age. Also display the total amount to be paid in a day as salary to the employees by the company. Assume appropriate member variables and member functions.
24. Write the prefix and postfix form of the expression $(A + B) / C - D$.
25. Write a program to evaluate the equation, $A = B * C$ using classes and objects where A, B and C are objects of the same class.
26. What is a collision? How will you resolve the collisions, while inserting {89, 18, 49, 58, 69} into the hash table using various methods in open addressing hash table?
27. Create a base class named, 'shape' with two members base and height, a member function for initialization and a virtual function to compute area(). Derive two specific classes Triangle and Rectangle which override the function area(). Use these classes in the main function and display the area of a triangle and a rectangle using virtual function.