



**Question bank for**  
**EC6301 – Object Oriented Programming and Data Structures**

**SEM/YEAR/DEPT : III / II /ECE**

**PREPARED BY :T.A.VINAYAGAM/AP/CSE**

**UNIT I**  
**PART A**

- 1. What is the output of the following program, if it is correct? Otherwise indicate the mistake: [May 2006]**

```
int l=10; Void
main []
{ int l=20;
  { int l=30;
    cout<<l<<endl;
  }}
}
```

The program is in correct due to syntax error. Within the main function, there is no need of another opening braces in the int l=20; and also closing braces.

- 2. Difference between Class and structure? [April -2010, Dec-2012]**

- Class is the ADT where as structure is udt.
- Class needs access specifier such as private, public & private where as structure members can be accessed by public by default & don't need any access specifiers.
- Class is oops where structure is borrowed from traditional structured [pop] concept.

- 3. What is abstract Class? [Nov-2009]**

An abstract class is a class that is designed to be specifically used as a base class. An abstract class contains at least one pure virtual function. You declare a pure virtual function by using a pure specifier [= 0] in the declaration of a virtual member function in the class declaration

- 4. List out the advantages of new operator over malloc[] [Dec-2012]**

- It automatically computes the size of the data object. It
- automatically returns the correct pointer type
- It is possible to initialize the objects while creating\_ the memory space. It
- can be overloaded.

- 5. What are the basic concepts of OOS? [ April -2011]**

Objects.  
Classes.  
Data abstraction and Encapsulation.  
Inheritance.  
Polymorphism.  
Dynamic binding.  
Message passing

- 6. What is the difference between local variable and data member? [Nov-2011]**

- A data member belongs to an object of a class whereas local variable belongs to its current scope.

- A local variable is declared within the body of a function and can be used only from the point at which it is declared to the immediately following closing brace.
- A data member is declared in a class definition, but not in the body of any of the class member functions.
- Data members are accessible to all member function of the class.

### **7. What is the function parameter? Difference between parameter and Argument. [Nov-2011]**

A **function parameter** is a variable declared in the prototype or declaration of a function:

```
void foo[int x]; // prototype -- x is a
parameter void foo[int x] // declaration -- x
is a parameter
{
}
```

An **argument** is the value that is passed to the function in place of a parameter

### **8. What is data hiding? [April -2011,Nov-2010]**

The insulation of data from direct access by the program is called as data hiding or information binding.

The data is not accessible to the outside world and only those functions, which are wrapped in the class, can access it.

### **9. What are the advantages of Default Arguments? [Nov-2010]**

The function assigns a default value to the parameter which does not have a matching argument in the function call. They are useful in situations where some arguments always have the same value.

e.g., float amt [float P, float n, float r = 0.15];

### **10. List out the basic concepts of Object Oriented Programming. [Nov-2009]**

- Objects
- Classes
- Data Abstraction and Encapsulation
- Inheritance
- Polymorphism
- Dynamic Binding
- Message Passing
- 

### **11. What are abstract classes? [No2009, Apr 2013]**

Classes containing at least one pure virtual function become abstract classes. Classes inheriting abstract classes must redefine the pure virtual functions; otherwise the derived classes also will become abstract. Abstract classes cannot be instantiated.

### **12. Define abstraction and Encapsulation [Apr2011] Data Abstraction**

Abstraction refers to the act of representing the essential features without including the background details or explanations.

#### **Data Encapsulation**

The wrapping up of data and functions into a single unit is known as data encapsulation.

### **13. What is the Need for Static Members [April 2011]**

Class members can be declared using the storage class specifier static in the class member list. Only one copy of the static member is shared by all objects of a class in a program. When you declare an object of a class having a static member, the static member is not part of the class object.

**14. Define Polymorphism. [Apr 2013]**

Polymorphism is another important oops concept. Polymorphism means the ability to take more than one form. For example, an operation may exhibit **different behavior in different instances**. Behavior depends upon the types of data used in the operation.

**15. What do you mean by pure virtual functions? [Dec2008]**

A pure virtual member function is a member function that the base class forces derived classes to provide. Any class containing any pure virtual function cannot be used to create object of its own type.

**16. Write a C++ program to check the given integer is prime or composite number. [Apr-2010]**

```
#include<conio.h>
#include<stdio.h>
int main[]
{
    int num,d,ctr;
    clrscr();
    printf("\n Enter a number=");
    scanf("%d",&num);
    d=1;    ctr=0;
    while[d<=num]
    {
        if[num%d==0]
            ctr++;d=d+1;
    }
    if[ctr==2]
        printf("\n %d is a prime number",num);
    else
        printf("\n %d is a composite number",num);
    getch();
    return[0];
}
```

**17. What is function Prototype? [DEC 2011]**

A function prototype or function interface in C, Perl, PHP or C++ is a declaration of a function that omits the function body but does specify the function's return type, name and argument types. While a function definition specifies what a function does, a function prototype can be thought of as specifying its interface.

**18. List out four Storage Classes in C++ [Nov 2008]**

Storage classes are used to specify the lifetime and scope of variables. How storage is allocated for variables and how variable is treated by compiler depends on these storage classes. These are basically divided into 5 different types :

1. Global variables
2. Local variables
3. Register variables
4. Static variables
5. Extern variables

**19. What is an identifier?**

Identifiers are names for various programming elements in c++ program. such as variables, arrays, function, structures, union, labels ect., An identifier can be Composed only of uppercase, lower case letter, underscore and digits, but should start only with an alphabet or an underscore.

**20. What is a keyword?**

Keywords are word whose meanings have been already defined in the c compiler. They are also called as reserved words.

(ex) main(), if, else, else, if, scanf, printf, switch, for, goto, while ect.,

**21. List out the benefits of oops.**

- Can create new programs faster because we can reuse code
  - Easier to create new data types
  - Easier memory management
  - Programs should be less bug-prone, as it uses a stricter syntax and type checking.
  - 'Data hiding', the usage of data by one program part while other program parts cannot access the data Will whiten your teeth
- 22. List out the application of oops.**
- Client server computing
  - Simulation such as flight simulations.
  - Object-oriented database applications.
  - Artificial intelligence and expert system
  - Computer aided design and manufacturing systems.

**23. Define data hiding.**

The purpose of the exception handling mechanism is to provide a means to detect and report an “exceptional circumstance” so that appropriate action can be taken.

**24. What is the use of scope resolution operator?**

In C, the global version of the variable cannot be accessed from within the inner block. C++ resolves this problem by introducing a new operator :: called the scope resolution operator. It is used to uncover a hidden variable.

Syntax:

:: variable name

**25. When will you make a function inline?**

When the function definition is small, we can make that function an inline function and we can mainly go for inline function to eliminate the cost of calls to small functions.

**26. What is overloading?**

Overloading refers to the use of the same thing for different purposes.

There are 2 types of overloading:

- Function overloading
- Operator overloading

**27. What is the difference between normal function and a recursive function? A**

recursive function is a function, which call it whereas a normal function does not.

Recursive function can't be directly invoked by main function

**28. What are objects? How are they created?**

Objects are basic run-time entities in an object-oriented programming system. The class variables are known as objects. Objects are created by using the syntax:

classname obj1,obj2,...,objn;

(or) during definition of the class:

class classname

{

-----

}obj1,obj2,...,objn;

**29. List some of the special properties of constructor function.**

- They should be declared in the public section.
- They are invoked automatically when the objects are created.
- They do not have return types, not even void and cannot return values.
- Constructors cannot be virtual.

Like other C++ functions, they can have default arguments

**30. Describe the importance of destructor.**

A destructor destroys the objects that have been created by a constructor upon exit from the program or block to release memory space for future use. It is a member function whose name is the same as the class name but is preceded by a tilde. Syntax:

```
~classname(){ }
```

**31. What do you mean by friend functions?**

C++ allows some common functions to be made friendly with any number of classes, thereby allowing the function to have access to the private data of these classes. Such a function need not be a member of any of these classes. Such common functions are called friend functions.

**32. What are member functions?**

Functions that are declared within the class definition are referred as member function.

**33. Define dynamic binding.**

Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time.

**34. Write any four properties of constructor. (DEC 2010)**

- Constructors should be declared in the public section.
- They are invoked automatically when the objects are created.
- They do not have return types
- They cannot be inherited

**35. List any four Operators that cannot be overloaded. (DEC 2010) (DEC 2009) (DEC 2011)**

- Class member access operator ( . , .\* )
- Scope resolution operator (::)
- Size operator ( sizeof )
- Conditional operator (?:)

**36. What is a Destructor? (DEC 2012)**

A destructor is used to destroy the objects that have been created by a constructor. It is a special member function whose name is same as the class and is preceded by a tilde '~' symbol. When an object goes out from object creation, automatically destructor will be executed.

**Example:**

```
class File {  
public:  
~File(); //destructor declaration  
};  
File::~File()  
close(); // destructor definition  
  
}
```

### 37.What is the Need for initialization of object using Constructor? (DEC 2012)

If we fails to create a constructor for a class, then the compiler will create a constructor by default in the name of class name without having any arguments at the time of compilation and provides the initial values to its data members. So we have to initialize the objects using constructor

### 38.What is a Copy Constructor (DEC 2009)

A copy constructor is used to declare and initialize an object from another object. It takes a reference to an object of the same class as an argument

Eg: integer i2(i1);

would define the object i2 at the same time initialize it to the values of i1.

Another form of this statement is

Eg: integer i2=i1;

The process of initializing through a copy constructor is known as copy initialization.

### 39.Give an example for a Copy Constructor (JUNE 2013)

```
#include<iostream>
#include<conio.h>
using namespace std;
class Example    {
    // Variable Declaration
    int a,b;
    public:
    //Constructor with Argument
    Example(int x,int y)    {
    // Assign Values In Constructor
    a=x;
    b=y;
    cout<<"\nIm Constructor";
    }
    void Display()    {

    cout<<"\nValues : "<<a<<"\t"<<b;
    }
};
int main()    {
    Example Object(10,20);
    //Copy Constructor
    Example Object2=Object;
    // Constructor invoked.
    Object.Display();
    Object2.Display();
    // Wait For Output Screen
    getch();
    return 0;
```

### 40.What is the Need for Destructors? (June 2013)

- Destructor is used to destroy an object.
- By destroying the object memory occupied by the object is released.

**41.Explain the functions of Default Constructor(MAY 2011)**

- The main function of the constructor is, if the programmer fails to create a constructor for a class, then the compiler will create a constructor by default in the name of class name without having any arguments at the time of compilation and provides the initial values to its data members. Since it is created by the compiler by default, the no argument constructor is called as default constructor.

**42.What is the need for Overloading an operator(MAY 2011)**

To define a new relation task to an operator, we must specify what it means in relation to the class to which the operator is applied. This is done with the help of a special function called operator function. It allows the developer to program using notation closer to the target domain and allow user types to look like types built into the language. The ability to tell the compiler how to perform a certain operation when its corresponding operator is used on one or more variables.

**43.What is the function of get and put function (MAY 2010)**

- `Cin.get(ch)` reads a character from `cin` and stores what is read in `ch`.  
`Cout.put(ch)` reads a character and writes to `cout`

**Part B**

1. Illustrate the working of constructors and destructors with an example.
2. Explain the control structures of C++ with suitable examples.
3. Illustrate the use of copy constructors and function overloading with C++ programming.
4. Specify a class called `complex` to represent complex numbers. Overload `+`, `-` and `*` operators when working on the object of this class.
5. Define friend function specify its importance. Explain with suitable example.
6. What is parameterized constructor? Explain with example.
7. What is operator overloading? Overload the numerical operators `+` and `/` for complex numbers “addition and division” respectively.
8. Write short notes on container classes and proxy classes.
9. Write short notes on Dynamic Memory Allocation and Static Class Members.
10. Briefly explain about oops concepts.

## UNIT II

### PART A

#### 1. What are templates? (AUC DEC 2009)

- Template is one of the features added to C++.
- It is new concepts which enable us to define generic classes and functions and thus provides support for generic programming.
- A template can be used to create a family of classes or functions.

#### 2. Illustrate the exception handling mechanism. (AUC DEC 2009)

- C++ exception handling mechanism is basically upon three keywords, namely, **try, throw, and catch.**
- **try** – is used to preface a block of statements which may generate exceptions.
- **throw** - When an exception is detected, it is thrown using a throw statement in the try block.
- **Catch**- A catch block defined by the keyword catch 'catches' the exception 'thrown' by the thrown statement in the try block and handles its appropriately.

#### 3. What happens when a raised exception is not caught by catch block? (AUC MAY 2010)

- If the type of object thrown matches the arg type in the catch statement, then catch block is executed for handling the exception.
- If they do not match the program is aborted with the help of abort() function which is invoked by default.
- When no exception is detected and thrown, the control goes to the statement immediately after the catch block. That is the catch block is skipped.

#### 4. Give the syntax of a pointer to a function which returns an integer and takes arguments one of integer type and 2 of float type. What is the difference between a class and a structure?

int (X\_fun) (int, float);(AUC MAY 2010)

#### 5. What is a template? What are their advantages (AUC DEC 2010/JUN 2012/DEC2012)

- A template can be used to create a family of classes or functions.
- A template is defined with a parameter that would be replaced by a specified data type at the time of actual use of the class or functions.
- Supporting different data types in a single framework.
- The template are sometimes called parameterized classes or functions.

**6. How is an exception handled in C++? (AUC DEC 2010)**

Exceptions are run time anomalies or unusual conditions that a program may encounter while executing.

1. Find the problem (Hit the exception)
2. Inform that an error has occurred.( Throw the exception)
3. Receive the error information. ( Catch the exception)
4. Take corrective actions.( Handle the exception)

**7. What is function template? Explain.**

**(AUC MAY 2011)**

- Like class template, we can also define function templates that could be used to create a family of functions with different argument types.
- The general format of a function template is:

```
template<class T>

returntypefunctionname ( arguments of type T)

{

// .....

//.....Body of function with type T wherever appropriate

//.....

}
```

**8. List five common examples of exceptions.**

**(AUC MAY 2011)**

- Division by zero
- Access to an array outside of its bounds
- Running out of memory or disk space.

**9. What is class template? (AUC DEC 2011)**

- Templates allows to define generic classes. It is a simple process to create a generic class using a template with an anonymous type.
- The general format of class template is:

```
template<class T>
```

```
class classname
```

```
{// .....class member specification.
```

```
//..... with anonymous type T wherever appropriate
```

```
//.....
```

```
}
```

**10. What are 3 basic keywords of exception handling mechanism?  
(AUC DEC 2011)**

C++ exception handling mechanism is basically built upon three keywords

- try
- throw
- catch

**11. What are the c++ operators that cannot be overloaded?**

Size operator (sizeof)

Scope resolution operator (::)

member access operators( . , .\* )

Conditional operator (?:)

**12. What is a virtual base class?**

When a class is declared as virtual c++ takes care to see that only copy of that class is inherited, regardless of how many inheritance paths exist between the virtual base class and a derived class.

**13. What is the difference between base class and derived class?**

The biggest difference between the base class and the derived class is that the derived class contains the data members of both the base and its own data members. The other difference is based on the visibility modes of the data members.

**14. What are the rules governing the declaration of a class of multiple inheritance?**

- More than one class name should be specified after the : symbol.
- Visibility modes must be taken care of.

If several inheritance paths are employed for a single derived class the base class must be appropriately declared

**15. Mention the types of inheritance.**

1. Single inheritance.
2. Multiple inheritance.
3. Hierarchical inheritance.
4. Multilevel inheritance.
5. Hybrid inheritance.

**16. Define dynamic binding.****APRIL AMY 2010**

Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time.

**17. What do u mean by pure virtual functions?**

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 redeclare it as a pure virtual function. A class containing pure virtual functions cannot be used to declare any objects of its own.

**18. Mention the key words used in exception handling.**

The keywords used in exception handling are

- throw
- try
- catch

**19. List the ios format function.**

The ios format functions are as follows:

- width()
- precision()
- fill()
- setf()
- unsetf()

**20. List the manipulators.**

The manipulators are:

- setw()
- setprecision()
- setfill()
- setiosflags()
- resetiosflags()

**21. Mention the equivalent ios function for manipulators.**

Manipulator      Equivalent ios function

setw(int w)      width()

setprecision(int d)      precision()

setfill(int c)      fill()

setiosflags(long f)      setf()

resetiosflags(long f)      unsetf()

endl      "\n"

**22. Define fill functions.**

The fill( ) function can be used to fill the unused positions of the field by any desired character rather than by white spaces (by default). It is used in the following form: cout.fill(ch); where ch represents the character which is used for filling the unused positions. For example, the statements

```
cout.fill('*');
```

```
cout.width(10);
```

```
cout<<5250<<"\n";
```

will produce the following output:

**23 .Give the syntax of exception handling mechanism.**

The syntax of exception handling mechanism is as follows:

```
try
{
-----
throw exception
-----
}
catch(type arguments)
{
-----
-----
}

}
```

**PART B**

1. Explain in detail about virtual function and polymorphism with example.
2. Briefly explain this pointer with example program.
3. Explain the concept of virtual destructors.
4. What is inheritance? Explain with examples the different types of inheritance in C++
5. Write about casting class pointers and member function.
6. Write short notes on constructors and destructors in derived classes.
7. Briefly explain about abstract base classes and concrete classes.
8. Explain protected data with private and public inheritance.
9. Explain the concept of overriding. How it differs from overloading?
10. Write short notes on dynamic binding and protected members.

## UNIT III PART-A

### 1. Write down the definition of data structures?

NOV DEC 2012

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. Binary Heap

NOV DEC 2012 ,APRIL MAY 2009

The implementation we will use is known as a binary heap. Its use is so common for priority queue implementations that when the word heap is used without a qualifier

#### Structure Property

A heap is a binary tree that is completely filled, with the possible exception of the bottom level, which is filled from left to right. Such a tree is known as a complete binary tree

### 3. Define Algorithm?

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.

### 4. What are the features of an efficient algorithm?

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

### 5. List down any four applications of data structures?

Compiler design  
Operating System  
Database Management system  
Network analysis

### 6. What is meant by an abstract data type (ADT)?

An ADT is a set of operation. A useful tool for specifying the logical properties of a datatype is the abstract data type. ADT refers to the basic mathematical concept that defines the datatype. Eg. Objects such as list, set and graph along their operations can be viewed as ADT's.

### 7. What are the operations of ADT?

Union, Intersection, size, complement and find are the various operations of ADT. **8. What is meant by list ADT?**

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  $a_i$ .

### 9. What are the two parts of ADT?

- Value definition
- Operator definition

### 10. What is a Sequence?

A sequence is simply an ordered set of elements. A sequence  $S$  is sometimes written as the enumeration of its elements, such as  
 $S = \{a_1, a_2, \dots, a_n\}$   
If  $S$  contains  $n$  elements, then length of  $S$  is  $n$ .

**11. Define len(S),first(S),last(S),nilseq ?**

len(S) is the length of the sequence S.

first(S) returns the value of the first element of S

last(S) returns the value of the last element of S

nilseq :Sequence of length 0 is nilseq .ie., contains no element.

**12. What are the two basic operations that access an array?**

Extraction:

Extraction operation is a function that accepts an array, a ,an index,i,and returns an element of the array.

Storing:

Storing operation accepts an array , a ,an index i , and an element x.

**13. Define Structure?**

A Structure is a group of items in which each item is identified by its own identifier ,each of which is known as a member of the structure.

**14. Define Union ?**

Union is collection of Structures ,which permits a variable to be interpreted in several different ways.

**15. Define Automatic and External variables?**

Automatic variables are variables that are allocated storage when the function is invoked.

External variables are variables that are declared outside any function and are allocated storage at the point at which they are first encountered for the remainder of the program's execution.

**16. What is a Stack?**

**NOV DEC 2008**

A Stack is an ordered collection of items into which new items may be inserted and from which items may be deleted at one end, called the top of the stack. The other name of stack is Last-in -First-out list.

**17. What are the two operations of Stack?**

- \_ PUSH
- \_ POP

**18. What is a Queue?**

A Queue is an ordered collection of items from which items may be deleted at one end called the front of the queue and into which items may be inserted at the other end called rear of the queue.Queue is called as First-in-First- Out(FIFO).

**19. What is a Priority Queue?**

**NOV DEC 2010**

Priority queue is a data structure in which the intrinsic ordering of the elements does determine the results of its basic operations. Ascending and Descending priority queue are the two types of Priority queue.

**20. What is a linked list?**

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

**21. What is a doubly linked list?**

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.

**22. Define double circularly linked list?**

In a doubly linked list, if the last node or pointer of the list, point to the first element of the list, then it is a circularly linked list.

**23. What is a circular queue?**

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

**24. Define max and min heap?**

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 is called a min heap.

**PART B**

1. Explain about Linked list, its Types, insertion and deletion routines with suitable example.
2. Explain the insertion and deletion operation in singly linked list.
3. Explain array based implementation of list with an example program.
4. Given singly linked list whose first node is pointed to by the pointer variable C formulate an algorithm to delete the first occurrence of X from the list and Insert the element X after the position P in the list.
5. Explain the implementation stack using linked list.
6. Write the ADT operation for insertion and deletion routine in stack.
7. Explain the process of postfix, prefix, infix expression evaluation with an example.
8. Give a procedure to convert an infix expression  $a+b*c+(d*e+f)*g$  to postfix notation.
9. Write a routine to insert an element in a linked list
10. What is a queue? Write an algorithm to implement queue.
11. Explain the process of conversion from infix expression to postfix using stack.
12. Write the ADT operation for insertion and deletion routine in linked lists & Queue

## UNIT IV

### **PART A (2Marks)**

#### **1. Define non-linear data structure?**

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

#### **2. Define tree?**

A tree is a data structure, which represents hierarchical relationship between individual Data items.

#### **3. Define child and parent of a tree.**

The root of each subtree is said to be a child of 'r' and 'r' is the parent of each subtree root.

#### **4. Define leaf?**

In a directed tree any node which has out degree 0 is called a terminal node or a leaf.

#### **5. What is a Binary tree?**

A Binary tree is a finite set of elements that is either empty or is partitioned into three disjoint subsets. The first subset contains a single element called the root of the tree. The other two subsets are themselves binary trees called the left and right subtrees.

#### **6. What are the applications of binary tree?**

- Binary tree is used in data processing.
- a. File index schemes
- b. Hierarchical database management system

#### **7. What is meant by traversing?**

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

#### **8. What are the different types of traversing?**

- The different types of traversing are
- 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.

#### **9. What are the two methods of binary tree implementation?**

Two methods to implement a binary tree are,

- a. Linear representation.
- b. Linked representation

**10. Define Graph?**

A graph  $G$  consists 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 of edges  $E$  to a set of pairs of elements of  $V$ .

It can also be represented as  $G=(V, E)$ .

**11. Define adjacent nodes?**

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  $v$ .

**12. Name the different ways of representing a graph?**

- a. Adjacency matrix
- b. Adjacency list

**13. What are the two traversal strategies used in traversing a graph?**

- a. Breadth first search
- b. Depth first search

**14. What is an acyclic graph?**

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

**15. Give some example of NP complete problems.**

- Hamiltonian circuit.
- Travelling salesmen problems

**16. What are the algorithms used to find the minimum spanning tree?**

An AVL tree is a binary search tree with a balancing condition. For every node in the tree the height of the left and right subtrees can differ at most by 1. The height of

an empty tree is defined to be -1. It ensures that the depth of the tree is  $O(\log N)$

**17. What is topological sort?**

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

**18. What is single source shortest path problem?**

Given as an input a weighted graph,  $G=(V,E)$  and a distinguished vertex, 's' find the shortest weighted path from 's' to every other vertex in  $G$ .

**19. Mention some shortest –path problems.**

Unweighted shortest paths

Dijkstra's algorithm

All-pairs

shortest

paths

**21. Define complete binary tree.**

It is a complete binary tree only if all levels, except possibly the last level have the maximum number of nodes maximum.

A complete binary tree of height 'h' has between  $2^h$  and  $2^{h+1} - 1$  node

**22. Define binary search tree. Why it is preferred rather than the sorted linear array and linked list?**

Binary search tree is a binary tree in which key values of the left sub trees are lesser than the root value and the key values of the right sub tree are always greater than the root value.

In linear array or linked list the values are arranged in the form of increasing or decreasing order. If we want to access any element means, we have to traverse the entire list.

But if we use BST, the element to be accessed is greater or smaller than the root element means we can traverse either the right or left sub tree and can access the element irrespective of searching the entire tree.

**23. Give various implementations of trees.**

Linear implementation

Linked list implementation

**24. List out the application of trees.**

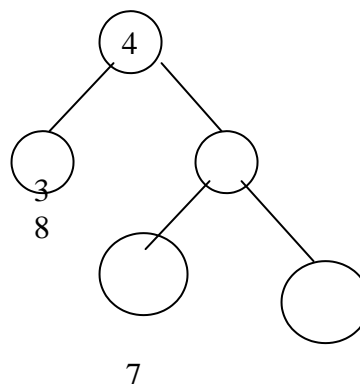
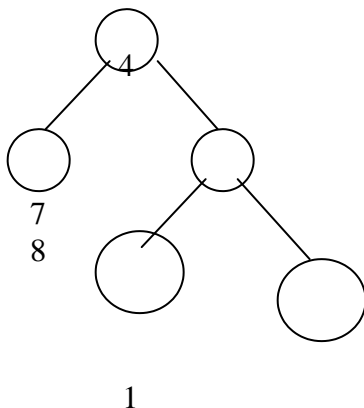
Ordered tree

“C” – representation of tree

**25. What is the difference between binary tree and binary search tree?**

Binary Tree	Binary Search Tree
A tree is said to be a binary tree if it	A BST is binary tree in which the key values in the left sub tree is lesser than the

has at most two children	root and key values of the right sub tree is greater than the root value.
It doesn't have any order.	It should be in order.



26. Show the maximum number of nodes in a binary tree of height  $H$  is  $2^{H+1} - 1$ .  
Consider  $H = 3$

No. of nodes in a full binary tree =  $2^{H+1} - 1$

$$= 2^{3+1} - 1$$

$$= 2^4 - 1$$

$$= 15 \text{ nodes}$$

We know that a full binary tree with height  $h=3$  has maximum 15 nodes.  
Hence proved.

27. List out the cases involved in deleting a node from a binary search tree.

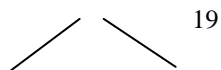
Case 1: Node to be deleted is a Leaf node

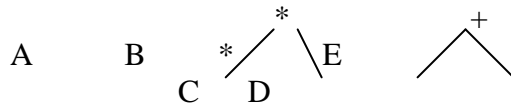
Case 2: Node to be deleted has one child

Case:3: Node to be deleted has two children.

28. Write the pre-order, in-order, post-order traversal for the tree. (AUC NOV/ DEC 2010)

+





In order traversal:  $(A * B) + (C * D) + E$

Pre order traversal:  $+ * A B + * C D E$  Post order traversal:  $A B * C D * E + +$

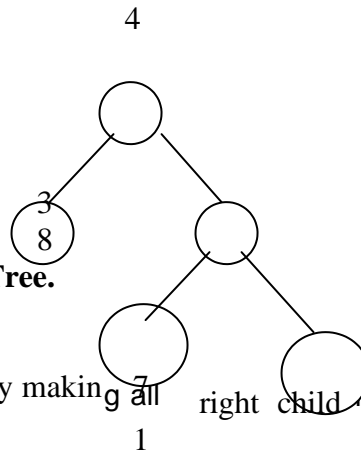
**29.  $A + (B - C) * D + (E * F)$ , if the above arithmetic expression is represented using a binary tree, Find the number of non-leaf nodes in the tree.**

Expression tree is a binary tree in which non – leaf nodes are operators and the leaf nodes are operands.

In the above example, we have 5 operators. Therefore the number of non-leaf nodes in the tree is 5.

**30. What is a BST – binary search tree? (AUC NOV/DEC 2009)**

Binary search tree is a binary tree in which key values of the left sub trees are lesser than the root value and the key values of the right sub tree are always greater than the root value.



**31. Define threaded Binary Tree. (AUC NOV/DEC 2010)**

A binary tree is threaded by making all right child pointers that would normally

be null point to the inorder successor of the node, and all left child pointers that would normally be null point to the inorder predecessor of the node.

**32. Write the advantages of threaded binary tree.**

**(AUC NOV/ DEC2009)1.**

By doing threading we avoid the recursive method of traversing a Tree , which makes use of stack and consumes a lot of memory and time.

2. The node can keep record of its root

**33. Write an algorithm to declare nodes of a tree structure.**

**(AUC APR / MAY**

**2010)**

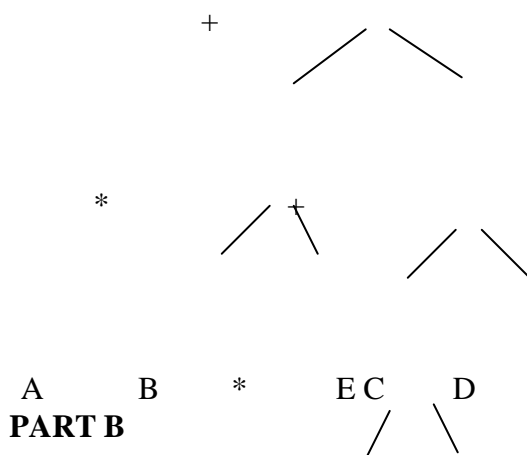
Struct tree node

```
{
int    element; Struct tree
*left; Struct tree *right;
}
```

**34. Give an example for Expression Trees.**

**(AUC APR / MAY**

**2010)**



**PART B**

1. Explain the operations of insertion of nodes into and deletion of nodes from, a binary search tree with code.
2. Narrate insertion deletion with example.
3. Give the analysis of insertion and deletion operations of nodes in binary search tree.
4. Draw the binary search tree for the following input list 60, 25,75,15,50,66,33,44. Trace an algorithm to delete the nodes 25, 75, 44 from the tree.
5. Explain the operations of insertion of nodes into and deletion of nodes from, a binary search tree with code.
6. Explain the two applications of trees with a neat example.
7. How do you insert an element in a binary search tree?
8. What are the graph traversal methods? Explain it with example.
9. Give an algorithm to find minimum Spanning tree, explain it with suitable example.
10. Write an algorithm for finding minimum spanning tree and explain application, illustrate the algorithm with typical data of your own example.
11. Explain the algorithm for depth first search and breadth first search with the following graph
12. Explain connected component.

## UNIT V

### PART A

#### 1. What is maxheap?

Apr/May,2010

If we want the elements in the more typical increasing sorted order, we can change the ordering property so that the parent has a larger key than the child. it is called max heap.

#### 2. What is divide and conquer strategy?

In divide and conquer strategy the given problem is divided into smaller problems and solved recursively. The conquering phase consists of patching together the answers. Divide and conquer is a very powerful use of recursion that we will see many times.

#### 3. Differentiate between merge sort and quick sort? Apr/May,2011

Mergesort

1. Divide and conquer strategy
2. Partition by position

Quicksort

1. Divide and conquer strategy
2. Partition by value

#### 4. Mention some methods for choosing the pivot element in quicksort?

1. Choosing first element
2. Generate random number
3. Median of three

#### 5. What are the three cases that arise during the left to right scan in quicksort?

- I and j cross each other
- I and j do not cross each other
- I and j points the same position

#### 6. What is the need of external sorting?

External sorting is required where the input is too large to fit into memory. So external sorting is necessary where the program is too large.

It is a basic external sorting in which there are two inputs and two outputs tapes.

#### 7. Define multi way merge?

Apr/May,2011

If we have extra tapes then we can expect to reduce the number of passes required to sort

#### 8. Define polyphase merge?

The k-way merging strategy requires the use of 2 k tapes. This could be prohibitive for some applications. It is possible to get

## 9. What is replacement selection?

We read as many records as possible and sort them. Writing the result to some tapes. This seems like the best approach possible until one realizes that as soon as the first record is written to a output tape the memory it used becomes available for another record. If the next record on the input tape is larger than the record we have just output then it can be included in the item. Using this we can give algorithm. This is called replacement selection.

## 10. What is sorting?

Sorting is the process of arranging the given items in a logical order. Sorting is an example where the analysis can be precisely performed.

## 11. What is mergesort?

The mergesort algorithm is a classic divide and conquer strategy. The problem is divided into two arrays and merged into single array

## 12. What are the properties involved in heapsort?

Apr/May,2010

1. Structure property
2. Heap order property

## 13. Define articulation points.

If a graph is not biconnected, the vertices whose removal would disconnect the graph are known as articulation points.

## 14. What is meant by Expression Tree? (Apr / May – 13)

An expression tree is a binary tree in which the operands are attached as leaf nodes and operators become the internal nodes.

### PART B

1. Sort the sequence 3, 1, 4,7,5,9,2,6,5 using Insertion sort.
2. Explain the operation and implementation of merge sort.
3. Explain the operation and implementation of external sorting.
4. Write quick sort algorithm and explain.
5. Trace the quick sort algorithm for the following list of numbers. 90,77,60,99,55,88,66
6. Write down the merge sort algorithm and give its worst case, best case and average case analysis.
7. Explain the Quick sort algorithm with example
8. Briefly explain about quick sort algorithm
9. Trace the quick sort algorithm for the following numbers 90,77,60,99,55,88,66
10. Explain linear search & binary search algorithm in detail.
11. Explain linear search algorithm with an example.
12. Briefly differentiate linear search algorithm with binary search algorithm.