DHANALAKSHMI SRINIVASAN INSTITUTE OF RESEARCH AND TECHNOLOGY
SIRUVACHUR, PERAMBALUR – 611 113

DEPARTMEN OF COMPUTER SCIENCE AND ENGINEERING

CS6301 - PROGRAMMING AND DATA STRUCTURES

QUESTION BANK

UNIT I

1. Define object oriented programming.

Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating such modules on demand.

2. List out the features of OOPS .

- Classes
- Objects
- Data abstraction
- Data encapsulation
- Inheritance
- Polymorphism
- Message passing
- Extensibility
- Persistence
- Delegation
- Genericity
- Dynamic binding.

3. **What is an object ?**

Objects are the basic run time entities in an object oriented system. They may represent a person , a place or any data item. Objects contain data and code to manipulate data.

4. What is a class?

The entire set of data and code of an object that can be made a user defined data type with the help of a class. A class is a collection of objects of type class.

5. What is data abstraction?

The technique of creating new data types that are well suited to an application to be programmed is known as data abstraction.

6. What is data encapsulation?

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

7. What is data hiding?

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

8. What are inheritance / reusability / derivation?

Inheritance is the process by which objects of one class acquire the properties of objects of another class.

The concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it .

Derivation involves the creation of new classes (derived class) from the existing ones (base class).

9. What is polymorphism?

Polymorphism means the ability to take more than one form.

10. What is the use of a break statement?

A break construct terminates the execution of loop and the control is transferred to the statement immediately following the loop. The term break refers to the act of breaking out of a block of code.

11. What is the use of a continue statement?

The continue statement skips the remainder of the current iteration initiates the execution of the next iteration.

12. Define function.

The process of splitting a large program into small manageable tasks ane designing them independently is popularly called modular programming or divide and conquer technique. A repeated group of instructions in a program can be organized as a function. A function is a set of program statements that can be proceesed independently.

13. What is a pointer? What are the uses of a pointer?

Pointer is defined as a variable used to store memory addresses.

Uses :

- Accessing array elements .
- Passing arguments to a function when the function needs to modify the original.
- Passing arrays and strings to functions.
- Creating data structures such as linked lists, binary tree etc .
- Obtaining memory from the system dynamically.

14. What is a friend function?

Friend function is a special type of function which is used to access all the private and protected members of a class. The functions that are declared with the keyword friend are called friend functions. A function can be a friend to multiple classes.

15. What are the properties of a friend function?

- A friend function is not in the scope of the class to which it has been declared as friend.
- 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 with each member name.

16. What is the difference between friend function and member function?

The only difference between a friend function and member function is that, the friend function requires the argument to be explicitly passed to the function and processes them explicitly, whereas, the member function considers the first argument implicitly.

17. What is an inline function?

Inline functions are those whose function body is inserted in place of the function call statement during the compilation process. With the inline code the program will not incur any context switching overhead.

18 .What is a recursive function?

A function that contains a function call to itself or a function call to a second function which eventually calls the first function is known as recursive functions.

19. What is data conversion?

When we use the assignment operator we assign a value on the right hand side to a variable on the left side & if it is of a different data type then we have to perform data conversion or type conversion.

20. What are the methods of data conversion?

There are two methods for data conversion:

(i) implicit data conversion (ii) explicit data conversion

21. Give the structure of a C++ program.

Include files
Class definition
Member function definitions
Main function

22. What is function prototype?

Function prototype is otherwise known as function declaration. The prototype describes the function interface to the compiler by giving details such as the number and type of arguments and the type of return values.

23. What is a class ? How will you define a class?

A class is a way to bind the data and its associated functions together. A class specification has two parts :

- Class declaration
- Class function definition

24. What are the characteristics of a static data member?

- Static data member is initialized to zero when the first object of its class is created. No other initialization is permitted.
- Only one copy of that member is created for the entire class and is shared by all the objects of that class, no matter how many objects are created.
- It is visible only within the class, but its life time is the entire program.

25. What are the properties of a static member function?

- A static 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;

26. In what way is a private member function different from public member function.

A private member function can only be called by another function that is a member of its class. Even an object cannot invoke a private function using the dot operator.

UNIT II

1.What is a constructor ?

A constructor is a special member function whose task is to initialize the objects of its class. It is special because its name is the same as the class name.

The constructor is invoked whenever an object of its associated class is created. It is called constructor because it constructs the values f data members of the class.

2. How will you declare and define a constructor

Constructor declaration :

```
Class integer
 {
    int m,n;
      public :
                    integer ( void )
 }
```

constructor definition

```
integer :: integer (void )
{
         m=0;
         n=0;
}
```

3. what are the characteristics of a constructor ?

- 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 therefore, they cannot return values.
- They cannot be inherited, though a derived class can call the base class constructor.
- Constructors cannot be virtual.

4. what are the types of a constructor ?

- Default constructor
- Parameterized constructor.
- Copy constructor

5. What is a parameteized constructor ?

The constructors that can take arguments are called parameterized constructors.

```
Integer ( int x, int y )
{
          m=x;
     n=y;
}
```

6. What is a default constructor ?

A constructor that accepts no parameter is called default constructor.

```
Default ( )

{
}
```

7. What is a destructor ?

A destructor as the name implies is used to destroy the objects that have been created by a constructor. Like a constructor the destructor is a member function whose name is the same as the class name but is preceded by a tilde symbol.

```
~ integer ( )

  {

   }
```

8. what is Dynamic constructors

The constructors can also be used to allocate memory while creating objects. This will enable the system to allocate the right amount of memory for each object when the objects are not of the same size thus resulting in the saving of memory.

Allocation of memory to objects at the time of their construction is known as dynamic constructors. The memory is allocated with the help of the new operator.

9. What is function overloading? Give an example.

Function overloading means we can use the same function name to create functions that perform a variety of different tasks.
Eg: An overloaded add ( ) function handles different data types as shown below.

// Declarations

i. int add(int a, int b); //add function with 2 arguments of same type

ii. int add(int a, int b, int c); //add function with 3 arguments of same type
iii. double add(int p, double q); //add function with 2 arguments of different type

10. What is operator overloading?

C++ has the ability to provide the operators with a special meaning for a data type. This mechanism of giving such special meanings to an operator is known as Operator overloading. It provides a flexible option for the creation of new definitions for C++ operators.

11. List out the operators that cannot be overloaded.

_ Class member access operator (. , .*)

_ Scope resolution operator (::)

_ Size operator (sizeof)

_ Conditional operator (?:)

13. What is the purpose of using operator function? Write its syntax.

To define an additional task to an operator, we must specify what it means in

relation to the class to which the operator is applied. This is done by Operator function , which describes the task. Operator functions are either member functions or friend functions. The general form is

```
return type classname :: operator (op-arglist )
{
function body
}
```

where *return type* is the type of value returned by specified operation.

Op-operator being overloaded. The *op* is preceded by a keyword operator. operator op is the function name.

14. Write at least four rules for Operator overloading.

_ Only the existing operators can be overloaded.
_ The overloaded operator must have at least one operand that is of user defined data type.
_ The basic meaning of the operator should not be changed.
_ Overloaded operators follow the syntax rules of the original operators. They cannot be overridden.

15. How will you overload Unary & Binary operator using member functions?

When unary operators are overloaded using member functions it takes no explicit arguments and return no explicit values.

When binary operators are overloaded using member functions, it takes one explicit argument. Also the left hand side operand must be an object of the relevant class.

16. How will you overload Unary and Binary operator using Friend functions?

When unary operators are overloaded using friend function, it takes one reference argument (object of the relevant class)
When binary operators are overloaded using friend function, it takes two explicit arguments.

17. How an overloaded operator can be invoked using Friend functions?

In case of unary operators, overloaded operator can be invoked as

Operator op (x);

In case of binary operators, overloaded operator can be invoked as

Operator op (x , y)

18. List out the operators that cannot be overloaded using Friend function.

_ Assignment operator =
_ Function call operator ( )
_ Subscripting operator [ ]
→_ Class member access operator

19. Explain basic to class type conversion with an example.

Conversion from basic data type to class type can be done in destination class. Using constructors does it. Constructor takes a single argument whose type is to be converted.
Eg: Converting int type to class type

```
class time
{

int hrs,mins;

public:
………….
Time ( int t) //constructor
{
hours= t/60 ; //t in minutes
mins =t % 60;
}
};
```

Constructor will be called automatically while creating objects so that this conversion is done automatically.

20. Explain class to basic type conversion with an example.

Using Type Casting operator, conversion from class to basic type conversion can be done. It is done in the source class itself.
Eg: vector : : operator double( )

```
{
double sum=0;

for(int I=0;I<size;I++)
sum=sum+v[ i ] *u[ i ] ;
return sqrt ( sum ) ;
}
```

This function converts a vector to the corresponding scalar magnitude.

21. Define inheritance ?

Inheritance is the process of creating new classes from the existing classes. The new classes are called derived classes. The existing classes are called base classes. The derived classes inherit all the properties of the base class plus its own properties. The properties of inheritance does not affect the base classes.

22. What are the types of inheritance ?

- Single inheritance
- Multiple inheritance
- Hierarchial inheritance
- Multilevel inheritance.
- Hybrid inheritance
- Multipath inheritance

23. What are the advantages of using a derived class ?

- New classes can be derived by the user from the existing classes without any modification.
- It saves time and money.
- It reduces program coding time.
- It increases the reliability of the program.

24.Define
multiple inheritance

If a derived class is derived from more than one base class, then it is called multiple inheritance.

hierarchial inheritance

If two or more derived classes are derived from the same base class then it is known as hierarchial inheritance

multilevel inheritance

If a derived class is derived from another derived class then it is known as multilevel inheritance.

25. Define abstract class

A class is said to be an abstract class if it satisfies the following conditions :

- It should act as a base class
- It should not be used to create any objects

26. What is an virtual function ?

A member function whose definition can be changed during run time is called virtual function. The class which contains virtual function is called polymorphic class and it should be a base class.

Different versions for the virtual function should be present in different derived classes with same name as virtual function name

27. Define pure virtual function ?

Pure virtual function is a virtual function with no body. The general form is

Virtual void member-function-name( ) = 0

28. What are the properties of pure virtual function ?

- It has no definition in the base class.
- We cannot create object for the class containing pure virtual function.
- Pure virtual function can be called through derived class objects.
- It acts as an empty bucket which has to be filled by its derived classes.

29. What are the rules for virtual functions ?

- Virtual functions must be members of some class.
- They cannot be static members.
- They are accessed by using object pointers.
- A virtual function can be a friend of another class.
- We can have virtual destructors, but we cannot have virtual constructors.

30. Define Polymorphism

It is the property of the same object to behave differently in different contexts given the same message. A single function name can be used for various purposes and single operator is used to achieve multiple operations and the usage of either the function at the operator depends on the context in such cases.

31. Define Compile time polymorphism:

The compiler while compiling the program resolves the function call or the operator call. This is known as compile time polymorphism

32. Define Runtime polymorphism

During multiple inheritances if the appropriate member function could be selected while the program is running is known as Runtime polymorphism

Unit III

1. What is an exception ?

Exception refers to unexpected condition in a program. The unusual conditions could be faults, causing an error which in turn causes the program to fail. The error handling mechanism of C++ is generally referred to as exception handling.

2. What are the types of exception ?

They are classified into synchronous and asynchronous exceptions.

Synchronous exception :

The exception which occur during program execution , due to some fault in the input data or technique that is not suitable to handle the current class of data, within the pgogram are known as synchronous exception.

For instance errors such as out-of-range, overflow, underflow and so on.

Asynchronous exception :

The exceptions caused by events or faults unrelated to the program and beyond the control of the program are called asynchronous exception.

For instance, errors such as keyboard interrupts, hardware malfunctions, disk failure, and so on.

3. What are the blocks related to exception handling constructs ?

The blocks related to exception handling constructs are

- try
- throw
- catch

The keyword try is used to preface a block of statements. Which may generate exceptions. This block of statements is known as try block.

When an exception is detected, it is thrown using throw statement in the try block.

A catch block catches the exception thrown by the throw statement in the try block and handles it appropriately.

Syntax:

```
try
{
    ………
………
throw exception;
}
catch (type arg)
{  }
```

4. What are the functions supported by C++ to handle uncaught exceptions ?

The functions supported by C++ to handle uncaught exceptions are

terminate ( )

set_terminate ( )

unexpected ( )

set_unexpected ( )

5.What is a template ?

Templates support generic programming , which allows to develop reusable software components such as functions, classes etc., supporting different data types in a single framework

6.What is function template ?

The templates declared for functions are called function templates. They perform appropriate operations depending on the data type of the parameters passed to them.

7. What is a class template ?

The templates declared for classes are called function templates. They perform appropriate operations depending on the data type of the parameters passed to them.

8. What is a stream ?

Stream is a series of bytes, which act either as a source from which input data can be extracted or as a destination to which the output can be sent. The source stream provides data to the program called te input stream and the destination stream that receives data from the program is called the output stream.

9. What are the types of standard streams ?

- cin - Standard input corresponding to stdin in C
- cout – Standard output corresponding to stdout in C
- cout – Standard error output corresponding to Stderr in C
- clog – A fully buffered version of cerr.

10. How do you classify ios class ?

Istream – input stream does formatted input.
Ostream – output stream does formatted output.
Iostream – input / output stream does formatted input and output.

11. What are manipulators ?

Manipulators are special functions that are specifically designed to modify the working of a stream. They can be embedded in the I/O statements to modify the form parameters of a stream.

12. Give few ios class functions and flags ?

Function	Task performed
Width( )	Specifies the required number of fields to be used while displaying the output value.
Precision ( )	Specifies the number of digits to be displayed after the decimal point.
Fill ( )	Specifies a character to be used to fill the unused area of a field. By default, fills blank space character.
Setf ( )	Sets format flag that control the form of output display.
Unsetf ( )	Clears the specified format flag.

13. What is a file ?

A file is a collection of related information defined by its creator. Commonly files represent programs (boyh source and object forms) and data. Files may be free-form, such as text files or may be rigidly formatted . In general, a file is a sequence of bits, bytes, lines, or records whose meaning is defined by its creator and user.

14. How many classes are used for handling files ?

ifstream – for handling input files.

ofstream – for handling output files.

fstream – for handling files on which both input and output can be performed.

15. What are the steps involved in manipulating a file ?

- Name the file on the disk
- Open the file to get the file pointer.
- Process the file. (Read / Write)
- Check for errors while processing.
- Close the file after its complete usage.

16. What functions are used for manipulation of file pointers ?

- seekg ( ) – Moves get file pointer to a specific location.
- seekp ( ) - Moves put file pointer to a specific location.
- tellg ( ) – Returns the current position of the get pointer
- tellp ( ) - Returns the current position of the put pointer

17. What do you mean by sequential access ?

A sequential file has to be accessed sequentially ; to access the particular data in the file all the preceding data items have to be read and discarded. For example a file on a tape must be accessed sequentially.

18. What do you mean by random access ?

A random file allows access to the specific data without the need for accessing its preceding data items. However, it can be accessed sequentially . For example, a file on a hard disk or floppy disk can be accessed either sequentially or randomly.

UNIT IV

PART A

1. Define tree.

Trees are non-liner data structure, which is used to store data items in a shorted sequence. It represents any hierarchical relationship between any data Item.

It is a collection of nodes, which has a distinguish node called the root and zero or more non-empty sub trees T1, T2,….Tk. each of which are connected by a directed edge from the root.

2. Define Height of tree.

The height of n is the length of the longest path from root to a leaf. Thus all leaves have height zero. The height of a tree is equal to a height of a root.

3. Define Depth of tree.

For any node n, the depth of n is the length of the unique path from the root to node n. Thus for a root the depth is always zero.

4. Define Degree of a node.

It is the number of sub trees of a node in a given tree.

5. Define Degree of a tree.

It is the maximum degree of a node in a given tree.

6. Define Terminal node or leaf?

Nodes with no children are known as leaves. A leaf will always have degree zero and is also called as terminal node.

7. Define Non-terminal node?

Any node except the root node whose degree is a non-zero value is called as a non-terminal node. Non-terminal nodes are the intermediate nodes in traversing the given tree from its root node to the terminal node.

8. Define sibling?

Nodes with the same parent are called siblings.

9. Define binary tree?

A Binary tree is a finite set of data items which is either empty or consists of a single item called root and two disjoin binary trees called left sub tree max degree of any node is two.

10. Define expression tree?

Expression tree is also a binary tree in which the leafs terminal nodes or operands and non-terminal intermediate nodes are operators used for traversal.

11. Define Construction of expression trees

a. Convert the given infix expression into postfix notation
b. Create a stack and read each character of the expression and push into the stack, if operands are encountered.

c. When an operator is encountered pop 2 values from the stack.

12.Define lazy deletion?

When an element is to be deleted.it is left in the tree itself and marked as being deleted. This is called as lazy deletion and is an efficient procedure if duplicate keys are present in the binary search tree, because the field that keeps count of the frequency of appearance of the element can be decremented of the element can be decremented.

13. Define AVL

AVL tree also called as height balanced tree .It is a height balanced tree in which every node will have a balancing factor of –1,0,1.

Balancing factor of a node is given by the difference between the height of the left sub tree and the height of the right sub tree.

14.What are the various operation performed in the binary search tree?

i. insertion
ii. deletion
iii. find
iv. find min
v. find max

15.What are the various transformation performed in AVL tree?

b. Single rotation
   1. Single L rotation
   2. Single R rotation
c. Double rotation
   1. LR rotation
   2. RL rotation

UNIT V

PART A

1. What is a graph?

A graph consists of a set of vertices V and set of edges E which is mathematically represented as G=(V,E).Each edge in a pair(V,W) where V,W,belongs to E ,edges are sometimes referred to as arcs.

2. What are Directed graphs?

If a pair of vertices for any edge is ordered, then that graph is called as Digraph or directed graph.

3. Define Path.

A path in a graph is a sequence of vertices w1,w2w,3,wN such that Wi,Wi+1 belongs to E for a value 1<=I<=N. The length of such a path is the number of edges on the path, which is equal to n-1.

4. Define Cycle.

A cycle is a path in which the first and last vertices are the same.

5. Define Acyclic graph.

A graph with no cycles is called Acyclic graph. A directed graph with no Edges is called as a directed Acyclic graph (or) DAG. DAGS are used for Compiler Optimization process.

6. Define Connected graph.

A graph is said to be a weighted graph is connected if there is a path from every vertex to every other vertex. A directed graph with this property is called as strongly connected graph. If a directed graph is not strongly connected but the underline graph. Without direction is connected it is called as a weakly connected graph.

7. What are the conditions for a graph to become a tree?

A graph is a tree if it has two properties.

i. If it is a connected graph.
ii. There should not be any cycles in the graph.

8. Define a Weighted Graphgraph if every edge in the graph is assigned

some weight or value. The weight of the edge is a positive value that represents the cost of moving the edge or the distance between two vertices.

9. Give the types of representation of graphs.

1. Adjacency matrix
2. Adjacency linked list

10.What is a minimum spanning tree?

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

11. Explain about Adjacency Matrix

Adjacency matrix consists of a n*n matrix where n is the no. of vertices present. In the graph, this consists of values either 0 or 1.

12. What is a back edge?

The possibility of reaching an already marked vertex is indicated by a dashed line, in a graph is called as back edge.

13. What is a single source shortest path problem?

Given as an input, a weighted graph, G=<V,E> and a distinguished vertex „S as the source vertex. Single source shortest path problem finds the shortest weighted path from s to every other vertex in G.

14. Explain about Unweighted shortest path.

Single source shortest path finds the shortest path from the source to each and every vertex present in a unweighted graph. Here no cost is associated with the edges connecting the vertices. Always unit cost is associated with each edge.

15. Explain about Weighted shortest path

Single source shortest path finds the shortest path from the source to each and every vertex present in a weighted graph. In a weighted graph some cost is always associated with the edges connecting the vertices.

16.What are the methods to solve minimum spanning tree?

a) Prim s algorithm
b) Kruskal s algorithm

17.Explain briefly about Prim s algorithm

Prim s algorithm creates the spanning tree in various stages. At each stage, a node is picked as the root and an edge is added and thus the associated vertex along with it.

18. Define a depth first spanning tree.

The tree that is formulated by depth first search on a graph is called as depth first spanning tree. The depth first spanning tree consists of tree edges and back edges.

19. What is a tree edge?

Traversal from one vertex to the next vertex in a graph is called as a tree edge.

****