OOPS VIVA QUESTIONS

UNIT-1

1. **What is OOP?**

Answer:- Object-Oriented Programming is a programming paradigm / model that organizes software design around data or objects rather than functions

2. **What is the difference between OOPS and POPS.**

Answer:- POPS follows top down approach while OOPS follows bottom up approach.

In POP program is divided into smaller parts called function, while in oops program is divided into smaller parts called objects.

3. **What are the features of OOPS?**

Answer:- Following are the 8 features of OOPS:-

I.) Objects:- An object is defined as an entity that contain data and its related function.

II.) Class:- A class is defined as a collection of objects with same type of data and functions.

III.) Data abstraction:- Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation.

IV.) Data encapsulation:- In Object Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulates them.

V.) Inheritance:- It is defined as sharing of attributes and functions among classes based on a hierarchical or sequential relationship.

VI.) Polymorphism:- A function is said to be polymorphic if it is applied to many different classes with different operation.

VII.) Dynamic Binding:- Binding is defined as the connection between the function call and its corresponding program code to be executed. In dynamic binding, the code to be executed in response to function call is decided at runtime.

VIII.) Message passing:- It is process of locating and executing a function in response to a message.

4. **Name some benefits of OOP.**

Answer:- 1. Data hiding is possible. 2. Reusablity is possible. 3. Overloading is possible

5. **What is method?**

Answer:- Methods are functions that belongs to the class.

6. **What is data types?**

Answer:- A data type is used to indicate the type of data value stored in a variable.

7. **What is primitive data type?**

Answer:- These data types are built-in or predefined data types and can be used directly by the user to declare variables. Example:- Integer, float, character, Boolean, etc

8. **What is derived data types?**

Answer:- The data-types that are derived from the primitive or built-in data types. Example:- array, pointer, etc

9. **What are user-defined data types?**

Answer:- The data types defined by the user are known as the user-defined data types. Example:- Structure, union, class, typedef,etc

10. **What is variable?** Answer:- A named memory location is called variable.
11. **What is constants?** **Answer:-** Constants are identifiers whose value does not change.
12. **What is the type of constants in c++?**

Answer:- Integer type, Floating point type, Character type, String type.

13. **What is pointer?**

Answer:- A pointer is a variable used to store the address of a memory cell.

14. **What is generic pointer?**

Answer:- Pointers of type void are known as generic pointers.

15. **What is references?**

Answer:- A special type of variable is called as enumerated or reference variable is used to provide alias (alternative name) for previously defined variable.

16. **What is typecasting?**

Answer:- Type conversion or typecasting of variables refers to changing a variable of one data type into another.

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

Answer:- In variables:- This is used to access global variables if same variables are declared as local and global.

In methods:- To define a member function outside the class definition we have to use the scope resolution :: operator along with class name and function name.

18. **What is access specifiers?**

Answer:- Access specifier or access modifiers are the labels that specify type of access given to members of a class. These are used for data hiding. These are also called as visibility modes.

19. **Type of access specifier:- private public protected**
20. **What is argument?**

Answer:- An argument is referred to the values that are passed within a function when the function is called. They are also called Actual Parameters

21. **What is parameter?**

Answer:- The values which are defined at the time of the function prototype or definition of the function are called as parameters. They are also called Formal Parameters.

22. What is UML?

Answer:- The Unified Modeling Language (UML) is the primary modeling language used to analyze, specify, and design software systems. UML diagrams can be classified into two groups: structure diagrams and behavior diagrams.

23. What is structure diagrams?

Answer:- These diagrams are used to show the static structure of elements in the system.

24. What does structure diagram include?

Answer:- The UML structure diagrams include

■Package diagram
■ Deployment diagram
■ Class diagram
■ Object diagram
■ Component diagram
■ Composite structure diagram

25. **What is behavior diagrams?**

Answer:- Those digram in which the dynamic behavioral semantics of a problem or its implementation is depicited

26. What does UML behavior digram includes?

Answer:- The UML Behavior diagrams include

■Use case diagram
■ Interaction diagrams
■ Activity diagram
■ Sequence diagram
■ State machine diagram
, etc

27. What is Package diagram?

Answer:- Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages.

28. What is class diagram?

Answer:- A class diagram is a view of the static structure of a system. It contains Packages, classes, interfaces, and relationships.

29. What is component diagram?

Answer:- Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems.

30. What is Deployment diagram?

Answer:- A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system.

31. What is use-case diagram?

Answer:- A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how).

32. What is Activity diagram?

Answer:- An activity diagram is a behavioral diagram that portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

33. What is implementation Diagram?

Answer:- It shows the implementation phase of systems development, such as the source code structure and the run-time implementation structure.

34. What are the types of implementation diagram?

Answer:- There are two types of implementation diagrams:

1. component diagrams
2. deployment diagrams

35. What is component diagram?

Answer:- Component diagrams are used in modeling the physical aspects of object-oriented systems.

36. What is deployment diagram?

Answer:- Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.

UNIT- 2

1. What is constructor?

Answer:- A constructor is a special type of member function of a class which initializes objects of a class.

2. What is destructor?

Answer:- A destructor, 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.   Eg: ~item() { }

3. What are the types of constructor?

Answer:- Constructors are of three types

- Default Constructor
- Parameterized Constructor
- Copy Constructor Object

4. What is overloading?

Answer:- It allows multiple definitions of a function with the same name, but different signatures

5. **What is default constructor?**

Answer:- Default constructor is the constructor which doesn’t take any argument. it has no parameters.

6. **What is parameterized constructor?**

Answer:- These are the constructors with parameter. Using this Constructor you can provide different values to data members of different objects, by passing the appropriate values as argument.

7. **What is Copy constructor?**

Answer:- These are special type of Constructors which takes an object as argument, and is used to copy values of data members of one object into other object. It creates a new object, which is exact copy of the existing copy, hence it is called copy constructor.

8. **Features of Polymorphism?**

Answer:- Polymorphism means the ability to take more than one form. An operation have different behaviour in different instances.

9. **What is constructor overloading/ multiple constructor?**

Answer:- It means having more constructor with the same name as class name.

10. **What are the types of polymorphism?**

Answer:- C++ supports 2 types of polymorphism, 1. Compile time Polymorphism 2. Run time Polymorphism.

11. **What is compile time polymorphism?**

Answer:- The overloaded functions are invoked by matching the type and number of arguments. This information is available at the compile time and, therefore, compiler selects the appropriate function at the compile time.

12. **What is run time polymorphism?**

Answer:- Run time polymorphism is achieved when the object's method is invoked at the run time instead of compile time.

13. **What is function overloading?**

Answer:- Function overloading is a feature of OOPs where two or more functions can have the same name but different parameters.

14. **What is operator overloading?**

Answer:- Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to the user-defined data type. It use **operator** keyword.

15. Operator that cannot be overloaded are as follows:

1. Scope operator (::)
2. Sizeof o member selector(.)
3. member pointer selector(*)
4. ternary operator(?:)

16. What are the types of operator overloading?

Answer:- Operator overloading types are

- Unary operator loading :- An unary operator means, an operator which works on single operand.
- Binary operator overloading :- An unary operator means, an operator which works on two operand.
- Operator Overloading using a friend function

UNIT- 3

1. What is inheritance?

Answer:- The capability of a class to derive properties and characteristics from another class is called Inheritance.

2. What are the types of inheritance?

Answer:-

- **Single Inheritance:** one derived class inherits from one base class.
- **Multiple Inheritance:** one derived class inherits from multiple base class(es)
- **Multilevel Inheritance:** wherein subclass acts as a base class for other classes. i.e a derived class is created from another derived class.
- **Hierarchical Inheritance**: wherein multiple subclasses inherited from one base class
- **Hybrid (Virtual) Inheritance:** Hybrid Inheritance is implemented by combining more than one type of inheritance. reflects any legal combination of other four types of inheritance

3. What is base class/ Super class?

Answer:- The class whose properties are inherited by sub class is called Base Class or Super class. Also known as parent class.

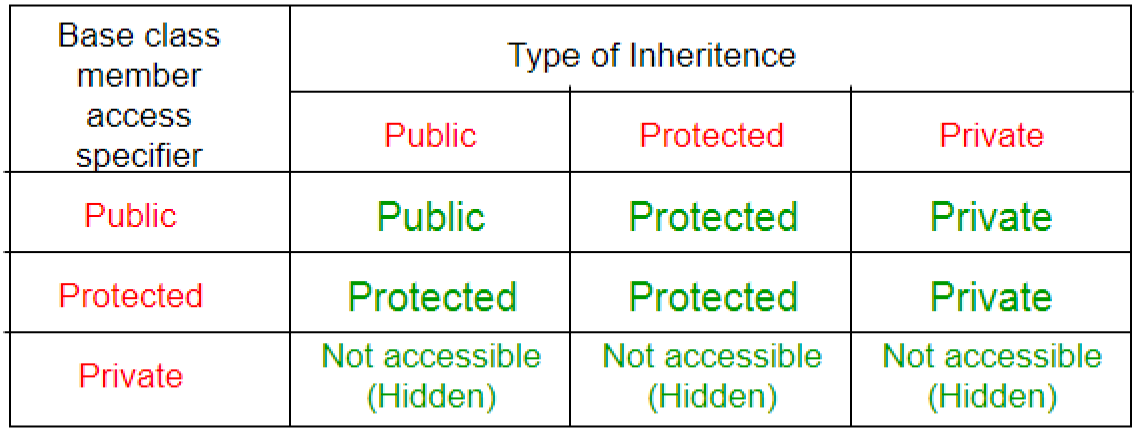
4. What is derived class/ Sub class?

Answer:- The class that inherits properties from another class is called Sub class or Derived Class. Also know as child class.

5. **What are the various mode of inheritance?**

Answer:- Following are the different mode of inheritance:-

1. Public mode
2. Private mode
3. Protected mode



Base class member access specifier	Type of Inheritence		
	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	Not accessible (Hidden)	Not accessible (Hidden)	Not accessible (Hidden)

6. **What is inline function?**

Answer:- Inline function is a function that is expanded in line when it is called. When the inline function is called whole code of the inline function gets inserted or substituted at the point of inline function call.

7. **What is friend function?**

Answer:- A friend function is a function that is declared outside a class, but is capable of accessing the private and protected members of class.

8. **What is the syntax of friend function?**

Answer:- friend ret_type func_name(arguments);

9. **What is friend class?**

Answer:- A friend function is a function that is declared outside a class, but is capable of accessing the private and protected members of class.

10. **What is function/method overriding?**

Answer:- If derived class defines same function as defined in its base class, it is known as function overriding in C++.

11. **What is virtual function?**

Answer:- A Virtual function is a member function in the base class whose definition is redefined in derived classes.

12. **Which keyword we use to make function virtual? Answer->> friend keyword.**
13. **What is Pure Virtual Function?**

Answer:- A pure virtual function (or abstract function) in C++ is a virtual function for which we can have implementation, But we must override that function in the derived class, otherwise the derived class will also become abstract class.

14. What is an abstract class?

Answer:- A class containing pure virtual functions cannot be used to declare any objects of its own. Such classes are called abstract base classes. **OR Abstract Class is a class which contains atleast one pure virtual function in it.**

UNIT- 4

1. What is a template?

Answer:- A template is a simple and yet very powerful tool in C++. The simple idea is to pass data type as a parameter so that we don't need to write the same code for different data types.

2. What are the types of templates?

Answer:- 1. Function template 2. Class template

3. Syntax of Templates:- template <typename T>
4. What is exception?

Answer:- Exceptions are run-time anomalies or abnormal conditions that a program encounters during its execution. Like we need 3 inputs and we get only 2 input, then there will be exception.

5. Name the keywords used for exception handling in c++.

Answer:- Try, catch, throw

6. What try keyword do?

Answer:- The try statement allows you to define a block of code to be tested for errors while it is being executed.

7. What throw keyword do?

Answer:- The throw keyword throws an exception when a problem is detected, which lets us create a custom error.

8. What catch do?

Answer:- The catch statement allows you to define a block of code to be executed, if an error occurs in the try blocks.

9. Which syntax we use when we don't know throw type?

Answer:- If we do not know the throw type used in the try block, you can use the "three dots" syntax (...) inside the catch block, which will handle any type of exception.

10. What is interface?

Answer:- Interfaces in C++ are nothing but a way to describe the behavior of a class without any commitment to any specific implementation of that class.

UNIT- 5

1. **What is STL?**

Answer:- The Standard Template Library (STL) is a set of C++ template classes to provide common programming data structures and functions such as lists, stacks, arrays, etc.

2. **What are the components of STL?**

Answer:- STL has four components:- 1. Algorithms 2. Containers 3. Functions 4. Iterators

3. **What is algorithms?**

Answer:- Algorithms in the STL are procedures that are applied to containers to process their data.

4. **What are the types of algorithm in c++ stl?**

Answer:- following are 5 types of algorithm in stl:-

1. Retrieve or Non-mutating Algorithms
2. Mutating Algorithms
3. Sorting Algorithms
4. Set Algorithms
5. Relational Algorithm

5. **What is containers?**

Answer:- Containers or container classes store objects and data.

6. **What is iterators?**

Answer:- Iterators are a generalization of the concept of pointers, they point to elements in a container

7. **What is functions?**

Answer:- The STL includes classes that overload the function call operator. Instances of such classes are called function objects or functors.

8. **What are the types of iterators?**

Answer:- there are 5 types of iterators:-

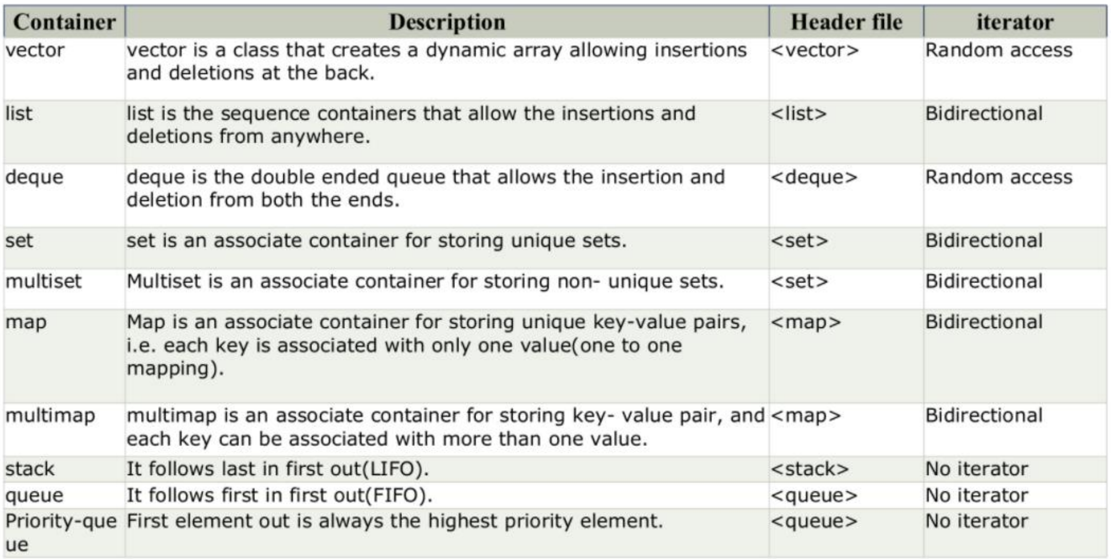
i.) Input iterator:
ii.) Output iterator
iii.) Forward iterator
iv.) Bi-directional iterator
v.) Random access iterator

9. **What are the type of containers?**

Answer:- following are the types of containers:-

1. Sequence containers:- Arrays, list, deque, vector
2. Associative containers:- set, map, multiset, multimap
3. Derived containers:- stack, queue, priority_queue

10.Commonly used containers:-



Container	Description	Header file	iterator
vector	vector is a class that creates a dynamic array allowing insertions and deletions at the back.	<vector>	Random access
list	list is the sequence containers that allow the insertions and deletions from anywhere.	<list>	Bidirectional
deque	deque is the double ended queue that allows the insertion and deletion from both the ends.	<deque>	Random access
set	set is an associate container for storing unique sets.	<set>	Bidirectional
multiset	Multiset is an associate container for storing non- unique sets.	<set>	Bidirectional
map	Map is an associate container for storing unique key-value pairs, i.e. each key is associated with only one value(one to one mapping).	<map>	Bidirectional
multimap	multimap is an associate container for storing key- value pair, and each key can be associated with more than one value.	<map>	Bidirectional
stack	It follows last in first out(LIFO).	<stack>	No iterator
queue	It follows first in first out(FIFO).	<queue>	No iterator
Priority-queue	First element out is always the highest priority element.	<queue>	No iterator

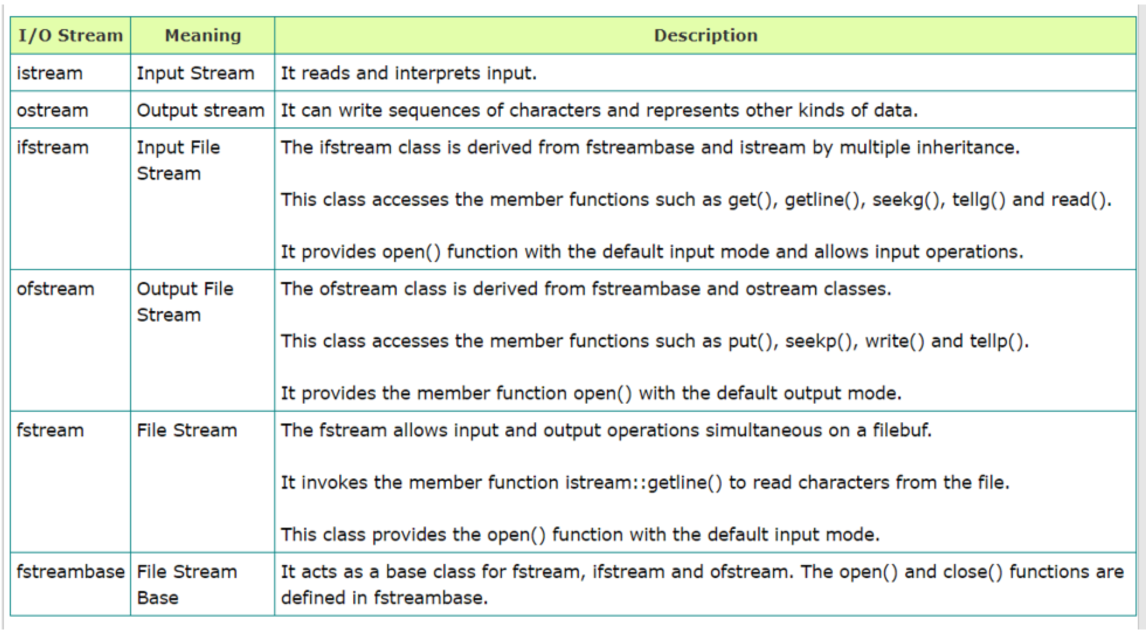
11. What is file?

Answer:- A file is a collection on information, usually stored on a computer’s disk. Information can be saved to files and then later reused.

12. What is stream?

Answer:- A transfer of information in the form of a sequence of bytes

13. Various types of streams:-



I/O Stream	Meaning	Description
istream	Input Stream	It reads and interprets input.
ostream	Output stream	It can write sequences of characters and represents other kinds of data.
ifstream	Input File Stream	The ifstream class is derived from fstreambase and istream by multiple inheritance. This class accesses the member functions such as get(), getline(), seekg(), tellg() and read(). It provides open() function with the default input mode and allows input operations.
ofstream	Output File Stream	The ofstream class is derived from fstreambase and ostream classes. This class accesses the member functions such as put(), seekp(), write() and tellp(). It provides the member function open() with the default output mode.
fstream	File Stream	The fstream allows input and output operations simultaneous on a filebuf. It invokes the member function istream::getline() to read characters from the file. This class provides the open() function with the default input mode.
fstreambase	File Stream Base	It acts as a base class for fstream, ifstream and ofstream. The open() and close() functions are defined in fstreambase.

Some list function:-

push_front(element) method is used to push elements into a list from the front.
push_back(element) method is used to push elements into a list from the back.
insert(iterator, element) : inserts element in the list before the position pointed by the iterator.
Empty() This method returns true if the list is empty else returns
Size() :- This method can be used to find the number of elements present in the list.
Reverse() :- This method can be used to reverse a list completely.
Sort() :- sort method sorts the given list

- The functions associated with stack are:

empty() – Returns whether the stack is empty
size() – Returns the size of the stack
top() – Returns a reference to the top most element of the stack
push(g) – Adds the element ‘g’ at the top of the stack
pop() – Deletes the top most element of the stack

SOME ALGORITHM:-

The find() algorithm looks for an element matching val between start and end
InputIterator find (InputIterator first, InputIterator last, const T& val);

count() returns the number of elements in the given range that are equal to given value.
count(first ,last ,value) :

The search() algorithm used to perform searches for a given sequence in a given range
search(first1 ,last1 ,first2 ,last2)

Sort() algorithm:- This function of the STL, sorts the contents of the given range
sort(start_iterator, end_iterator)

merge() :- Combines the elements in the sorted ranges [first1,last1) and [first2,last2), into a new range beginning at result with all its elements
OutputIterator merge (InputIterator1 first1, InputIterator1 last1, InputIterator2 first2, InputIterator2 last2, OutputIterator result);

The for_each() algorithm allows you to do something to every item in a container
Function for_each (InputIterator first, InputIterator last, Function fn);

The transform() algorithm does something to every item in a container, and places the resulting values in a different container (or the same one).
OutputIterator transform (InputIterator first1, InputIterator last1, OutputIterator result, UnaryOperation op);

NOTE:- THERE ARE LOTS AND LOTS OF FUNCTIONS FOR EACH CONTAINER TYPE IN STL(because it’s a huge topic by itself) . BUT I DON’T THINK THAT THESE FUNCTIONS TYPE OF QUESTION CAN BE ASKED IN VIVA.