OOPS

Question bank solution by **JAAT**

For more information about us visit https://youtu.be/aIC3suP4rZM

1.. Differentiate between data abstraction and data encapsulation.

Ans..

S.NO.	Abstraction
1.	Abstraction is the process or method of gaining the information.
2.	In abstraction, problems are solved at the design or interface level.
3.	Abstraction is the method of hiding the unwanted information.
4.	We can implement abstraction using abstract class and interfaces.
5.	In abstraction, implementation complexities are hidden using abstract classes and interfaces.
6.	The objects that help to perform abstraction are encapsulated.

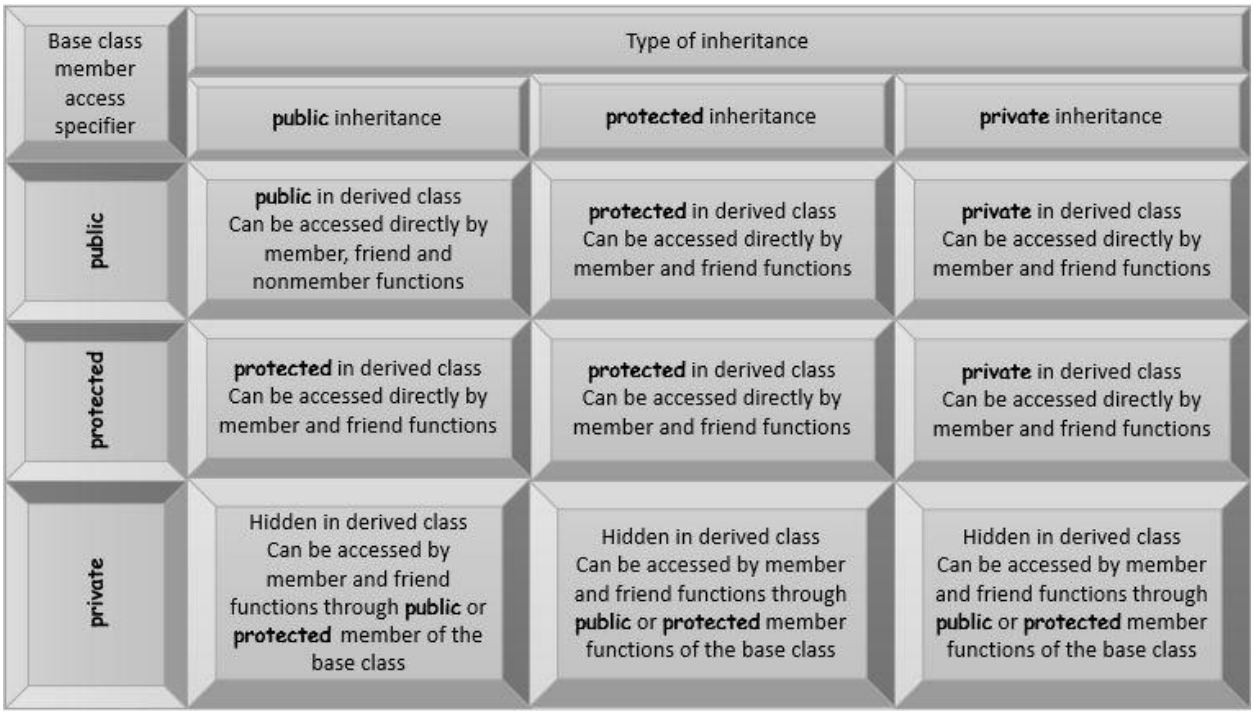
S.NO.	Encapsulation
1.	While encapsulation is the process or method to contain the information.
2.	While in encapsulation, problems are solved at the implementation level.
3.	Whereas encapsulation is a method to hide the data in a single entity or unit along with a method to protect information from outside.

4. Whereas encapsulation can be implemented using by access modifier i.e. private, protected and public.

5. While in encapsulation, the data is hidden using methods of getters and setters.

6. Whereas the objects that result in encapsulation need not be abstracted.

2.. Discuss the use of public, private and protected access specifiers and their visibility in the class.



3.. Discuss default constructor and parameterized constructor with the help of an example in C++.

ANS.. If no constructor is defined in the class then the compiler automatically creates one for the program. This constructor which is

created by the compiler when there is no user defined constructor and which doesn't take any parameters is called **default constructor**.

To put it up simply, the constructor that can take arguments are called parameterized constructor.

In practical programs, we often need to initialize the various data elements of the different object with different values when they are created. This can be achieved by passing the arguments to the constructor functions when the object is created.

4.. Write down the use of destructor in C++.

ANS.. Destructors in C++ are members functions in a class that delete an object. They are called when the class object goes out of scope such as when the function ends, the program ends, a delete variable is called etc.

5.. What is the need of constructor? How it is different from the member function?

ANS.. A constructor is a special type of member function of a class which initializes objects of a class. In C++, Constructor is automatically called when object(instance of class) is created. It is special member function of the class because it does not have any return type.

How constructors are different from a normal member function?

A constructor is different from normal functions in following ways:

- Constructor has same name as the class itself
- Constructors don't have input argument
- Constructors don't have return type
- A constructor is automatically called when an object is created.
- It must be placed in public section of class.

- If we do not specify a constructor, C++ compiler generates a default constructor for object (expects no parameters and has an empty body).

6 What is a static data member? How they are used in static functions? Explain with suitable illustrations

ANS.. Static data members are class members that are declared using static keywords. A static member has certain special characteristics. These are:

- 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 initialized before any object of this class is being created, even before main starts.
- It is visible only within the class, but its lifetime is the entire program

7 Define class and objects

ANS.. A class in C++ is the building block that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

8 What do you mean by dynamic binding? How it is useful in OOP?

ANS.. Dynamic binding also called **dynamic dispatch** is the process of linking procedure call to a specific sequence of code (method) at run-

time. It means that the code to be executed for a specific procedure call is not known until run-time. Dynamic binding is also known as *late binding* or *run-time binding*.

9.. Explain the use of friend function with the help of suitable example.

ANS.. If a function is defined as a friend function in C++, then the protected and private data of a class can be accessed using the function.

By using the keyword friend compiler knows the given function is a friend function.

For accessing the data, the declaration of a friend function should be done inside the body of a class starting with the keyword friend.

10. What is the need of overloading operators and functions?

ANS.. C++ allows you to specify more than one definition for a **function** name or an **operator** in the same scope, which is called **function overloading** and **operator overloading** respectively.

Function Overloading in C++

You can have multiple definitions for the same function name in the same scope. The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by return type.

Operators Overloading in C++

You can redefine or overload most of the built-in operators available in C++. Thus, a programmer can use operators with user-defined types as well.

Overloaded operators are functions with special names: the keyword "operator" followed by the symbol for the operator being defined. Like

any other function, an overloaded operator has a return type and a parameter list.

11 How do we invoke constructor? Can we have more than one constructor in a class? If yes, explain the need for such a situation

ANS Constructor function are invoked automatically when the objects are created.

Yes, we have when we need to overload the constructor, then we have to do this.

12 Write down the example to overload unary and binary operators in C++.

ANS..

13. State the use of scope resolution operator in C++.

In C++, scope resolution operator is **::**. It is used for following purposes.

1) To access a global variable when there is a local variable with same name:

2) To define a function outside a class.

3) To access a class's static variables.

4) In case of multiple Inheritance:

If same variable name exists in two ancestor classes, we can use scope resolution operator to distinguish.

5) For namespace

If a class having the same name exists inside two namespace we can use the namespace name with the scope resolution operator to refer that class without any conflicts

14 Compare and contrast the structured programming and object oriented programming.

Structured Programming

It is a subset of procedural programming.

Programs are divided into small programs or functions.

It is all about facilitating creation of programs with readable code and reusable components.

Its main aim is to improve and increase quality, clarity, and development time of computer program.

It simply focuses on functions and processes that usually work on data

It is a method of organizing, managing and coding programs that can give or provide much easier modification and understanding.

It generally follows “Top-Down Approach”.

It gives more importance of code.

Object-Oriented Programming

It relies on concept of objects that contain data and code.

Programs are divided into objects or entities.

It is all about creating objects that usually contain both functions and data.

Its main aim is to improve and increase both quality and productivity of system analysis and design.

In this, method works dynamically, make calls as per need of code for certain time.

It generally follows “Bottom-Up Approach”.

It gives more importance to data.

15. What is a dynamic constructor? Explain with suitable example.

ANS.. When allocation of memory is done dynamically using dynamic memory allocator new in a constructor, it is known as **dynamic constructor**. By using this, we can dynamically initialize the objects.

17. Compare and Contrast late binding and early binding

ANS

Early Binding

This is compile time polymorphism. Here it directly associates an address to the function call. For function overloading it is an example of early binding.

Late Binding

This is run time polymorphism. In this type of binding the compiler adds code that identifies the object type at runtime then matches the call with the right function definition. This is achieved by using virtual function.

18 What do you mean by implicit and explicit call of constructor? Explain with example.

Explicit Constructors

You may see warnings in certain C++ compilers about making certain constructors explicit.

Implicit Constructors

These constructors allow you to initialize a class value without specifying the name of the class.