OOPs Important Questions

1. What is Oops ?

It is a programming technique where everything revolves around the object.

2.What is Object ?

An object is an instance of a class, it is an entity which has two things, one is State or Properties , other one is Behavior or Method.

3.What is Class ?

A Class is a building block of Object Oriented Programming. To create user define data type we use Class. It contains the data members and member functions that operate on the data members. It is the blueprint of objects.
It can be two type -
a. Empty Class , b. Non-Empty Class

4.Why OOPs ?

1. Increase readability , it is reusable
2. Easy to maintain , easy to understand
3. We can relate with real life examples.
4. Useful for relatively big software.

5.Object Creation type ?

Object creation can be two type

1. Static Memory Allocation
2. Dynamic Memory Allocation

6. What is Access Modifier ?

Access Modifier define the scope of access. It is 3 types

1. Public
2. Private
3. Protected

7. What is Constructor ?

When we create an object at first “constructor” is called , Constructor helps to initialize the object.

a. It has no return type.
b. Name is same as Class
c. Initilize object.

8. Types of Constructor?

It has 3 types

1. Default Constructor
2. Parameterized Constructor
3. Copy Constructor

9. Copy Constructor !

In Copy Constructor we should do pass by reference bcz, if we do pass by value here, then we will stuck on infinite loop.

*** “this->” keyword refers to the current object

10. What is Destructor ?

Destructor is used for memory free , in static memory allocation destructor is automatically called but in dynamic we have to delete in manually.

a. it has no return type, no input parameter.

b. ~ sign in used in destructor.

c. when scope is end destructor is automatically called.

11. What are the main features / 4 Pillars of OOPs ?

1. Encapsulation
2. Inheritance
3. Polymorphism
4. Abstraction

12. What is Encapsulation?

Encapsulation is a such type of concept where we wrap the data , and hide it . It is the binding of data and methods that manipulate then into a single unit, where the sensitive data is hidden from the users. (Data Hiding)

Eg: Class Creation .

Shreya Pramanik

10. What is Perfect Encapsulation ?

If every data members are marked as private then that is called perfect Encapsulation.

11. Why Encapsulation ?

1. Increase security & privacy
2. Reusability
3. We can create read only data.

12. What is Inheritance?

Inheritance means Properties (Data members & Member functions) are inherited by Child Class from the Super Class.

Gfg: Inheritance means a class is derived from another class and uses data and implementation of that other class. The class which is derived is called child or derived or subclass and the class from which the child class is derived is called parent or base or superclass.

*** The main purpose of Inheritance is to increase code reusability. It is also used to achieve Runtime Polymorphism.

Access Modifier of Base Class	Mode of Inheritance		
	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	NA	NA	NA

*** We can not inherit private members at any mood .

Shreya Pramanik

13. Types of Inheritance ?

5 types

1. Single
2. Multi-Level
3. Multiple
4. Hierarchical
5. Hybrid

14. What is Dimond Problem?

If we take properties with same name in Multiple Inheritance then Dimond Problem happens.
that means same data member will present in Multiple Classes.

*** To solve Diamond Problem we use ‘Scope resolution Property’

15. What is Polymorphism ?

“Poly” means many , “morphism” means form. So , polymorphism means Existing in many forms.

Gfg : It is the property of some code to behave differently for different context.

16. Type of Polymorphism?

It is two types :

1. Compile Time Polymorphism
2. Run Time Polymorphism

17. Compile Time Polymorphism?

Compile time polymorphism, also known as static polymorphism , where the binding of the call to its code is done at the compile time.

It is based on two concept :

1. Function Overloading
2. Operator Overloading

18. Function Overloading means ?

When we create functions in same name , but we change either the number of the parameters , or the type of the parameters.

19. What is Run time Polymorphism?

Also known as dynamic polymorphism or late binding, When we start execution of the code , and if the polymorphism happened in run time , then that is called Run time Polymorphism. Function Overriding is happened here.

20. What is Function Overriding?

Function Overriding is a feature of Object Oriented Programming, that happened in Run Time Polymorphism, which allows the derived class to redefine a function of its base class with the help of virtual function. (virtual function will apply in parent class)

Advantages : Reusability, Custom Behavior

21. What is Abstraction ?

Abstraction means Implementation hiding. It means showing only the necessary information and hiding the other irrelevant information from the user.

Ig: Sort* a = new quicksort();