Day: Date:

INTERVIEW QUESTIONS

TECHNICAL PART :

Object Oriented Programming (OOP):

1) What is OOP?

OOP is a programming paradigm based on the concepts of objects, which can contain data and code.

- Data is in the form of fields
- Code is in the form of functions/procedures

2) Why we use OOP? Why we need OOP?

OOP is used to increase the reusibility and readibility of code. It makes the Code easy to use, debug and modify. It helps to keep the code "DRY" → 'Dont Repeat Yourself'.

3) Real Time Example of OOP?

Vehicle whose object is a car etc.
New, all the features of vehicle are in car and even in other objects

4) How many pillars of OOP and what?

There are four pillars of OOP

1) Encapsulation 2) Inheritence
3) Abstraction 4) Polymorphism

5) What is encapsulation and give a real time example.

- Is a technique of restricting a user from directly modifying the data member in order to maintain integrity of the data

→ Making the data member private

Real life Example:

Capsule is real life example of encapsulation As capsule binds all it's medicinal material.

6) What is Inheritance and give a real time example?

Is a technique of acquiring the properties of another class having features in common.

→ Inheriting the properties } IS-A Relationship

Real life Example:

- Animal class
- Dog object, Cat objects inherits the features of speak, watch, walk etc from Animal class.

7) What is Abstraction and give a real time example?

Is a technique of providing only the essential details to the user by hiding the unnecessary or irrelevant details of an entity.

```
→ class Car {
                  ↱ accessible to user
     public lock()
     private checkfuel() → Not accessible
  }
```

Real Time example:

Car, The user can drive, accelerate without knowing the actual working behined

8) What is polymorphism? real life example

polymorphism means different/many forms.
Its a feature that allows you to perform an action in multiple or different ways.

→ sum(int a, int b) sum(int a, int b, int c)

Real Life example:

A person can be a doctor, teacher, engineer
A man can be a son, father or brother.

9) What are the types of polymorphism

1) Static polymorphism (Method Overloading)
2) Dynamic polymorphism (Method Overriding)

10) What is static polymorphism?

- Also known as static binding or compile time binding.
- A type in which method calls are defined at the time of compilation.

11) What is method overloading?

Methods having same name but different datatypes or parameters.

sum(int a, int b) sum(double a, double b)

Sum is same but parameters are different.

Day: Date:

12) What is dynamic polymorphism ?

Also known as dynamic binding or run time binding.

- A type in which method calls are defined dynamically at the run time.
- It holds between two classes having IS-A relationship.

13) what is method Overriding ?

The methods having the same signature and also holds the inheritance b/w the classes.

14) What is static variable ? can we call a non static variable in static function.

- A variable that remains in memory while the program is running.
- It is initialized only once.
- It is shared among all the instances of a class

⇒ → Throw compile time error

No, non static variable can't be use in a static function while static variable can be used in both static and non static

15) What are the access modifiers?

There are three access modifiers

1-Private (Everything will be private in child)

2-Public (Only public part is public in child)

3-Protected (protected and public remains protected)

16) What is private class? can we make its object? where it can be used?

- A class that can be accessed within a certain file or directly.
- Yes, and no both are possible if we make constructor private then not possible and if public its possible.
- It can be used in nested classes to control the scope.

17) What is abstract class? object is possible?

- The class that can't be instantiated.
- The class having atleast one pure virtual function.
- The class in which atleast one function is declared but not defined.

⇒ we can't make its object but its possible to make its pointer.

18) What is interface?

It defines a behaviour that classes must implement. It is used to achieve Abstraction.

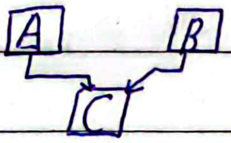
19) What are static classes?

- Classes that are used to hold utility methods and other functionalities that is not specific to particlilar instance of object.
- They can't be inherited.

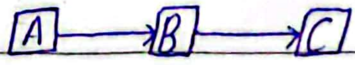
⇒ Math class is a static class. To use any method of class i-e Math.sqrt(), we do not need to create object.

20) Multiple and multilevel inherilence?

- In multiple inheritance each child can be derived from two or more parents



- In multiple inheritance, a class from which a class is derived is also derived from another class.



21) What is constructor:

- A type of function which is used to initialize the object
- It is called automatically when the object is created.
- Has no return type.
- Has no parameter

22) What is destructor?

- A type of function used to release the memory.
- Automatically called when the object is out of scope

23) Differentiate b/w shallow Copy and deep Copy?

- A shallow copy creates a new array but it doesn't create new copies of the elements within the array.
- It points to the address of original.
- Changes in copy will reflect in orignal
- A deep copy will create an independent copy of array and its data.
- Changes will not be reflected.

24) Is the size of int pointer & char pointer same?

Yes, as they only store the memory addresses and types are used to know What type of data they stored

25) Why copy constructor is used?

- It is used to initialize the members of a newly created object by copying the members of already existing object
- It is called when the object is created from

an existing object

26) Difference b/w pass by values & pass by reference.

The difference b/w pass-by-reference and pass by value is that modifications made to arguments passed in by reference in the called function have effect in the calling function, where as modifications made to arguments passed in by the value in the called function can not affect the calling function

⇒ **Pass-by-reference:** Refer the same memory variable as the caller.

⇒ **Pass-by-value:** Refer the copy of caller variable hence create two values in memory.

27) Your thoughts on Loop VS Recursion

Loop:

The process where the same set of instructions is repeated multiple times in a single cell.

Recursion:

The process where output of one iteration from a function call becomes the input of the next in a separate function call.

Day:________ Date:________

28) What is class and object?

Class: • A class is a user defined data type

- It contains data and methods
- It is a blueprint for creating object

Object: • Object is an instance of a class

- Object has the behaviour of class.

29) Differentiate b/w composition and aggregation?

Aggregation:

A relationship where the child can exist independently of the parent

Real-life example:

- A relationship b/w class (parent) and student (child)
- if class is removed, student can exists independently.
- It is **weak** relationship.
- It is **HAS-A** Relationship.

Composition:

A relationship where the child cannot exist independentlly of the parent.

Real-life example

- A relation b/w heart, lungs, kidney and body (parent)
- Is a **Strong** relationship
- It is **HAS-A** Relationship

30) where to use inheritance and where to use composition

where we have **IS-A** relationship we will use inheritance and where we will have **HAS-A** relationship, we will use composition.

Real-life Example:

Let we have 3 classes

1- Person 2-Student 3- Heart

So,

- Student is a person, So INHERITANCE
- Person has a heart, So COMPOSITION

31) What is a friend function

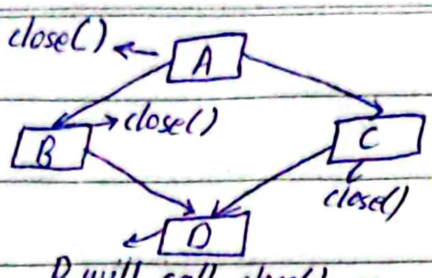
- A function that is friend of a given class.
- A function that is not a member of a class but can access the private and protected data of a given class.

32) What is Diamond problem? and how it can be resolved?

A problem that arises when a class inherits from two or more classes but have a common base class.

It will create multiple copies of base class.

Day: Date:



Now D will have multiple copies of A

D will call close(), so of which class it will call.

Solution

Make the parent classes (B, C) Virtual so there will be only one instance of base

33) What is virtual function?

A member function in the base class that we expect to redefine in derived class

Base → print() ← virtual
Derived → print() ← override
Base * b = & derived
b→print() → will print derive

34) What are disadvantages of OOP ?

It consumes more memory and CPU resources than other paradigms because objects store both data and methods so it requires more space and time to create and manipulate.

35) Limitations of OOP?

1- Larger problem size

2- Not suitable for all types of problems i-e small

3- It requires more data protection.

dynamic binding => virtual void fun() = 0 ;

Day: ________ Date: ________

31) Mostly asked binding question :

```
class base{                      class derived: public base{
  void fun(){                      void fun() {
  Iout<<"I am base";}                cout<<"I am child"
}                                }

 int main(){
  base* b = new derived();
    b→fun()
}
```

If dynamic binding, means base class is abstract or virtual, derived class fun will be called

If static binding, i-e no concept of overriding then base class fun will be called