

Interview Questions for CPP

Written by: Burhan Shaheen
student at International Islamic University
Islamabad

Question/Answers

1) When is Simple Class called Virtual class?

A Simple Class is Called Virtual Class when a Virtual Function is added as a member Function to Simple Class.

2) What is difference b/w simple and virtual class object. Briefly Explain with example.

Simple Class Object:

It can only hold its own object pointers.

This object contains simple, and no pointers are in this object.

Virtual Class Object:

This object stores its child pointers as well.

This Object are divided into two parts:

i. Data part

ii. Virtual pointer (points to Vtable)

3) What is Virtual table. Briefly explain with example.

To implement virtual functions, C++ uses a special form of late binding known as the virtual table. The virtual table is **a lookup table of functions used to resolve function calls in a dynamic/late binding manner**. A virtual table contains one entry for each virtual function that can be called by objects of the class.

In virtual table , There is address that points to current table and that table contain address of virtual function that points to code segments.

DD00 (this address is from virtual class object (Vptr))	
A100 (points to code segment virtual function)	
A200	
A300	

4)What is early and late binding?

Early Binding:

It calls a function at compile time and did not see the Content of the base class.

Late Binding:

It take place at run-time and this can be achieved by virtual function.

5)Why we use virtual function in polymorphism?

So If we does not use virtual with member function of base class in polymorphism then it did not see the content of derived class if we use base pointer for derived class.

6) What is abstract class and how we make simple class as abstract class?

Abstract class is a class that object is useless but their existence is important for derived class.” **If we want a base class to make them abstract we use a pure virtual function in base class after then, now if we made an object of base class, compiler shows an error.”**

Syntax: `virtual void fun_name () = 0;` //pure virtual fun

7) Why in virtual class we use virtual destructor?

A virtual constructor is not possible but virtual destructor is possible. Virtual Destructor are useful when you might potentially delete an instance of a derived class through the pointer of base class.

Class base

```
{  
    Public:  
    Virtual ~base () {}  
};
```

Class derived : public base

```
{  
    Public:  
    Virtual ~derived () {}  
};
```

Note : Destructor call is from bottom to top.

8) Write down the real-life example in which we apply polymorphism?

Mobile phone is a base class and abstract.

iPhone is derived class from Mobile.

Android is also derived class from Mobile.

Now we can't make object of Mobile because its use less. So, we will make object of derived class.

9) Why we use inheritance in our code or when?

It allows us to create a new class (derived class) from an existing class (base class). The derived class **inherits** the features from the base class and can have additional features of its own.

When we are going to specialize our Entity/class then we are using inheritance.

10) What is aggregation and composition, briefly explain with real life example?

Aggregation:  (“ has a “ relationship)

It implies the relationship where the child can exist independently of the parent.

Example:

Class (parent)

Student (child)

If we delete Class and the student Still Exist.

Composition:  (“ part of “ relationship)

It implies the relationship where the child can not exist independently of the parent.

Example:

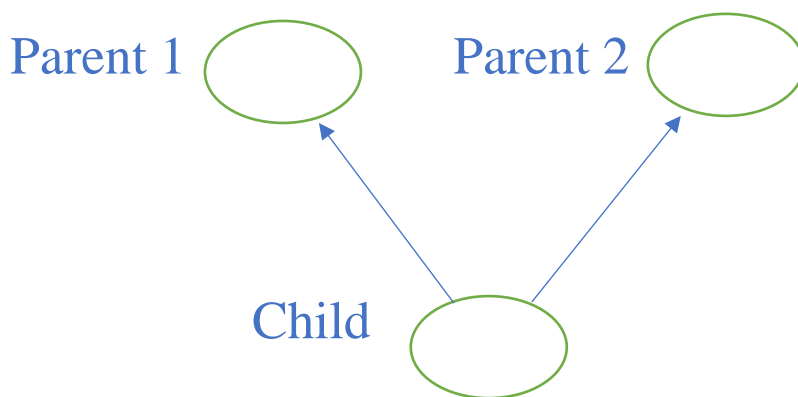
House (parent)

Room (child)

Room don't exist separate to a house.

11) What is multiple inheritance and what 2 problems occurs when we deal with multiple inheritance?

“A class may be derived from more than one base class”

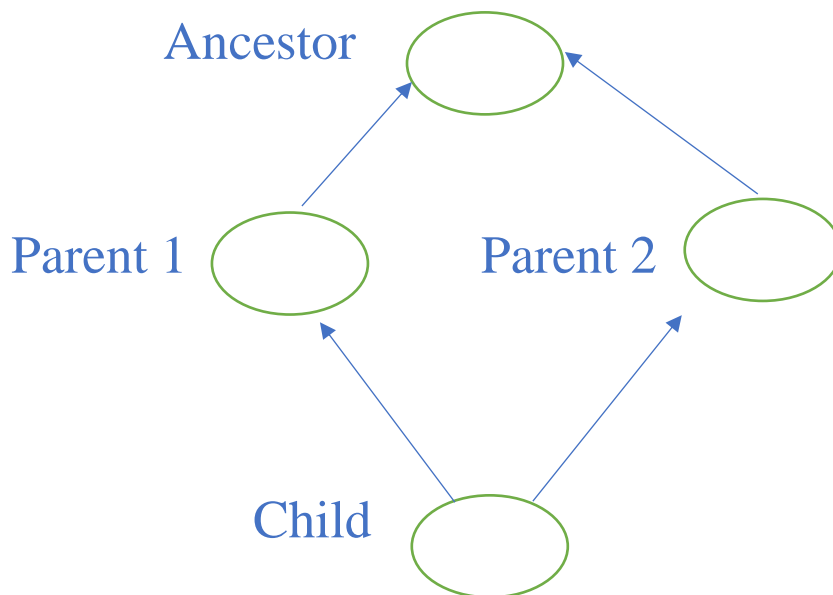


Problem1: Ambiguity

The ambiguity that arises when using multiple inheritance refers to **a derived class having more than one parent class that defines data members with the same name**. For example, if 'Child' inherits from both 'parent 1' and 'parent 2' and classes 'parent 1' and 'parent 2', both define a property named x and a

function named `getx()`. Then, which copy will 'child' get or will it get two copies, one from each parent?

Problem2: Repeated Base Class Problem



12) Why do we need to overload Operators in cpp?

Briefly explain with suitable example.

We need Operator Overloading in context where we want to apply Operator on User define data types.

Example:

If we have 2 objects of our class. O1, O2.

So, we want to apply addition on the (O1+O2) but we can't do this without Operator overloading.

13) What is Function overloading and overriding?

Function Overloading:

Within a single scope there may be more than multiple function having same name.

But how compiler will differentiate 😊

Through:

- i) No of input parameters
- ii) Order of parameter
- iii) Datatype of parameters

Function Over-riding:

When a Base class and Derived class contains a function with the same name and parameters then a derived class function hides(override) base class function. This is called Function Overriding.

14) What is difference between function invocation and function calling?

Function invocation:

When a Function that is not member of any class and calling of this function is called function invocation.

Function Calling:

When a Function that is member of class and called through object of class then this is called function calling.

15) What is prototype and when we use in our code?

Prototype is function header that is place before main class and body of this function header is place after main function. This improves readability of code.

16) Why we prefer inheritance then do changes in base class?

Suppose if we there are 5 class that is derived from base class. So, if all 5 classes want to change in base class then base class is messed up with a lot of changes that is made by every derived class. So that's why we prefer to do inheritance but not to change in base class.

17) What is constructor and destructor and what are types of them?

Constructor:

“Constructor is a special member function that construct or allocate memory and may initialize the data. It has no return type”

Types:

- 1) System Defined (initialize with garbage)
- 2) User Define (to initialize member by member)
- 3) Copy constructor (make copy of exiting object)

Destructor:

“Destructor is also a special member function that deallocate memory and is called automatically when object is going to destroy. Destructor is also called end function of program.”

Type:

- 1) System Define (its fine for static allocation)
- 2) User Define (when there is dynamic allocation like pointers then we use this)

18) Write down three paradigms from which you're familiar?

- a) Functional Paradigm
- b) Structured Paradigm
- c) Object Oriented Paradigm

19) What is the limitation of structured programming approach?

Limitation:

a) Unauthorized Access to Global data

(Function can access data of the structure easily)

b) Does not model the real world (best model is OOP)

c) Data is Undervalued (less importance given to data)

d) Not much extensible

(we cannot perform user friendly operations: $P3 = P2 - P1$)

20) Write down the principle of OOP?

Principle of OOP:

a) Encapsulation

b) Inheritance

c) Polymorphism

21) What are the advantages of encapsulation?

Advantages:

(i) Data Hiding

It hides the data from user.

(ii) Implementation Hiding

It Also hides implementation.

22) Write down two reasons for which we send parameter by reference?

Reason1:

We can return multiple values from function through parameter by Reference

Reason2:

When we send parameter by reference then the change made to that parameter values will also change to actual value.

THE END