# C/ C++ interview Practice

So to practice we are going to do two things – practice some interview questions for C/ C++ coders along with make a few pieces of code

<https://www.youtube.com/watch?v=6HuptuHyJZg>

<https://www.youtube.com/watch?v=1ONhXmQuWP8>

<https://www.interviewbit.com/cpp-interview-questions/>

<https://edabit.com/challenges/cpp>

Do a few hard challenges

Prepare for a C++ interview – Nail your C++ interview – online course – then become a c++ developer

## Object orientated programming

APIE – Abstraction, Polymorphism, Inheritance and Encapsulation

Definition of OOP – is a programming paradigm based on the concept of objects, contain data in form of attributes or properties, and actions in form of functions or methods. Basically taking real world objects and representing them within code.

E.g. computer monitor – data e.g. screen size, action e.g. turn on or off

### Abstraction

Only show necessary details to user about the object e.g. when user is turning computer on or off you don’t want to see the inner working.

Abstraction decouples user from coding.

### Inheritance

Code reusability

Lets say you have a class but want to make a new class but want to add some features to it. Classes derived from another class is called a sub class, extended class or child class.

Class from which others are derived called super class or parent class.

### Polymorphism

Most advanced topic

Poly – many

Morphism -forms

Determine what type of function to run whilst program is running.

Call methods on a class – if what’s calling it is a pointer of the parent object.

### Encapsulation

Data Hiding – restrict access to certain properties or actions in our object depending on who is trying to access/ call that object.

Private and public access – get and set functions .

Name space – does for functions and classes what scope does for variables. Allows you to use the same function or class name in different parts of the program without causing a name collision.

e.g. using namespace std allows you to remove std:: from before all the standard library functions in C++

## C++ Interview Questions

<https://www.interviewbit.com/cpp-interview-questions/>

C++ powerful and all purpose language developed by Bjarne Stroustrup at Bell Labs. Language is extension of C and one of fastest objected orientated programming languages – often chosen because high speed and compatibility.

### C++ questions for freshers

#### Difference between C and C ++

|  |  |
| --- | --- |
| **C** | **C++** |
| Procedure orientated programming language | Object orientated programming |
| Not support Data Hiding | Data hidden encapsulation to ensure data structures and operators are used as intended. |
| C is subset of C++ | C++ is superset of C |
| Function and operator overloading not supported by C | Function and operator overloading supported |
| Namespaces not present C | Namespaces be used to avoid collisions |
| Functions not be defined inside structures | Functions defined inside structures |
| Calloc() and malloc() functions used for memory allocation and free() is used for memory deallocation | New operator is used for memory allocation and deletes operator is used for memory deallocation. |

Class is user defined data type that has data members and functions. Data members are the data variables and members are functions used to perform operations of these variables.

Object is an instance of a class. Since class is user defined data type so object can also be called a variable of that data type.

#### Difference between struct and class

In C++ struct and class are the same except for a few differences in security. Difference:

|  |  |
| --- | --- |
| Structure | Class |
| Members structure public default | Members of class private by default |
| When deriving struct from class/ struct, default access specifiers for base class/ struct are public | Deriving a class, default access specifiers private |

#### What is operator overloading

Operator overloading essential element to perform operations on user defined types. Operator overloading modify the default meaning to operators such as +,-,\*

So basically you can add in a function to classes and overload them will values in new classes

#### Polymorphism

Two types of polymorphism – compile time and run time

Happens when we have multiple classes related to each other by inheritance

#### Constructor in C++

Member function that executed automatically whenever object created. Constructors have same name as class of which they are members so compiler knows that member function is a constructor. No return type is used for constructors.

#### Virtual Function

Member function in base class that you can redefine in a derived class. Virtual function is declared using virtual keyword. When function made virtual, C++ determines which function is to be invoked at runtime based on the type of the object pointed by the base class provider.

#### Difference between run time and compile time polymorphism

|  |  |
| --- | --- |
| Compile Time Polymorphism | Run Time Polymorphism |
| In this method, know at compile time which method called | We know at run time which method is called. Call not resolved by compiler. |
| Provides fast execution as we know at compile time | Slow execution compared to compile time. |
| Achieved by function overloading and operator overloading | Achieved virtual functions and pointers |

#### Friend class and friend function

Friend class access private, protected and public members of other classes in which it is declared as friends.

Like friend class, friend function can access private, protected and public members. Friend functions not member functions

#### C++ access specifiers

Public: All data members and member functions accessible outside class

Protected: All data and member functions are accessible inside class and to derived class.

Private: All data members and member functions not accessible outside class.

#### Data types present in C++

4 types:

1. Primitive (basic datatype) – char, short, int, float, long, doubl, bool
2. Derived – array, pointer
3. Enumeration – enum
4. User defined data types – structs, classes

#### Inline function

Function is inline - Compiler places copy that code of that function at each point where the function is called at compile time. One of important advantages using inline function is that eliminates the function calling overhead of traditional function. So rather than storing the function in memory then moving onto reference, execute code, get output, store output. You can just put the code where the function call is – useful for small code, requires recompiling.

#### References C++

Reference is like a pointer – another name of an already existing variable. Once reference name initialised with variable, variable can be accessed by reference name or variable name.

If change valuable of variable it is reflected – but if we reference a variable it cannot refer to any other variable. Can declare an array of pointers but array of references not possible.

#### Abstraction

Process showing essential details to user and hiding details which we don’t want to show to or user or hiding those irrelevant.

#### De-constructor overloading possible?

No de-constructor overloading not possible – de-constructors take no arguments, only one way to destroy object.

#### Call by value and call by reference

Call by value – pass copy parameter to functions. For these copied values a new memory is assigned and changes made to these values do not reflect the variable in the main function.

Call my reference we pass address of the variable and the address is used to access the actual argument used in the function call. Changes made in the parameter alter the passing argument.

#### What is an abstract class and when do you use it?

Class is called abstract class whose objects can never be created. Such class exists as a parent for a derived class. Make class abstract by placing pure virtual functions in class.

#### De-constructors in C

Constructor automatically called when object created. Similarly when object destroyed, function called destructor automatically called. Destructor same name constructor (which is same as class name) but it is preceded by a tilde ~. Used to deallocate memory.

#### Static Members and Static Member Functions

Variable in class declared static, space for it is allocated for lifetime of program. No matter how many objects class created there is only one copy of the static member. So same static member can be accessed by all objects of that class.

Static member function can be called even if no object of the class exist and the static functions are accessed by all objects of that class. Static member function can be called even if no objects class exist and the static function are accessed using only the class name and the scope resolution operator.

#### Inheritance

Inheritance is process creating a new class (derived, sub, extended) from existing class (parent or super class). Existing classes called base classes. Derived class has all cpaabilites of base class but can add new features and refinements of there own.

Permits code reusability.

### More advanced Interview Questions

#### Copy Constructor

Copy constructor member function that initialises an object using another object same class.

Define copy constructor. Don’t define the copy constructor default copy constructor called.

#### Difference between Shallow and Deep Copy

|  |  |
| --- | --- |
| Shallow Copy | Deep Copy |
| Stores references objects to original memory address | Makes new entry and separate copy of entire object with unique memory address. |
| Faster | Deep copy slower |
| Reflects changes made to new/ copied object in the original object. | Doesn’t reflect changes made new/ copied object in the original object |

#### Difference virtual functions and pure virtual functions

Virtual function = member function in base class that redefine in derived class. Declared using virtual keyword.

Pure virtual function is function no implementation and declared by assigning 0. 0 doesn’t reference anything it just tells compiler that it has no body. Forces derived class to instantiate function

#### Order of constructors called

Class D derived base class B.

Constructors are called in phases – most base class at top of inheritance tree called first. Each child class constructed in order until most child class constructed.

So during the destruction most child like class is called i.e. the most unlike the parent class.

#### Call virtual function constructor

IF call virtual function constructor behaviour different. Virtual call resolved run time. Always the member function of the current class that gets called and the virtual machine doesn’t work within the constructor.

#### Void Pointers

Pointer no datatype assigned to it. Can hold addresses any type.

Cannot turn pointer into another type unless you type cast it.

#### This Pointer in C++

Member functions every object pointer named this, points to object itself. Value this set to the address of the object which it is called.

#### Allocate and Deallocate memory

New operator used to allocate memory and deletes operator used for memory deallocation

Questions at the bottom of the page:

:: cannot be overloaded - it’s the cope declarer

Abstract class is a class that has at least one pure function

Size of void is 0

Memory used by an array is contiguous i.e. consecutive blocks of memory

To get the size of an array we need to do sizeof(arr)/sizeof(arr[0]) this is just byte logic – get the total size of the array in memory and then divide it by the size of one value in memory.

Note when passing for a function

so with an array of items - size must be passed from where the data is defined - so requires pre processing

when passed to the function it gets passed as a pointer so doing byte logic would return 0

Really funny error I came across recently – if you don’t but I=0 in a for loop – causes a segmentation fault if you are trying to

### Queues

Queues can be used with functions such as isempty(), pop(), first(), last(), empty()

### Threading

A thread is a block of code that executes – whenever working with multiple threads you are working with a multi threaded application. It means you can utilise the processor to execute code at the same time – this can be advantageous for speed.

Thread.join() means you wait to thread to finish and join the main code before moving on in main.

A mutex is an object that can be used to restrict access to certain shared data values to stop corruption. Can be locked and unlocked using a lockguard or the unlock and lock.

Remember to compile for c code with threads you need to include -std=c++11

g++ \*cpp -o messageApp -std=c++11

### Getting system Time

In C++11 we can call using chrono::system::time.now() and then covert it to a time\_t which is a time specific variable – therefore require ctime package to convert it from epoch to a readable time.

Linked Lists