Answers

Next Page »

Dear Readers, Welcome to **C++ Interview questions with answers** and explanation. These 73 solved **C++ Programming questions** will help you prepare for technical interviews and online selection tests during campus placement for freshers and job interviews for professionals.

After reading these tricky C++ questions, you can easily attempt the objective type and multiple choice type questions on **C++ Programming**.

Explain abstraction.

- Simplified view of an object in user's language is called abstraction.
- It is the simplest, well-defined interface to an object in OO and C++ that provides all the expected features and services to the user in a safe and predictable manner.
- It provides all the information that the user requires.
- Good domain knowledge is important for effective abstraction.
- It separates specifications from implementation & keeps the code simpler and more stable.

What is the real purpose of class – to export data?

No, the real purpose of a class is not to export data. Rather, it is to provide services. Class provides a way to abstract behaviour rather than just encapsulating the bits.

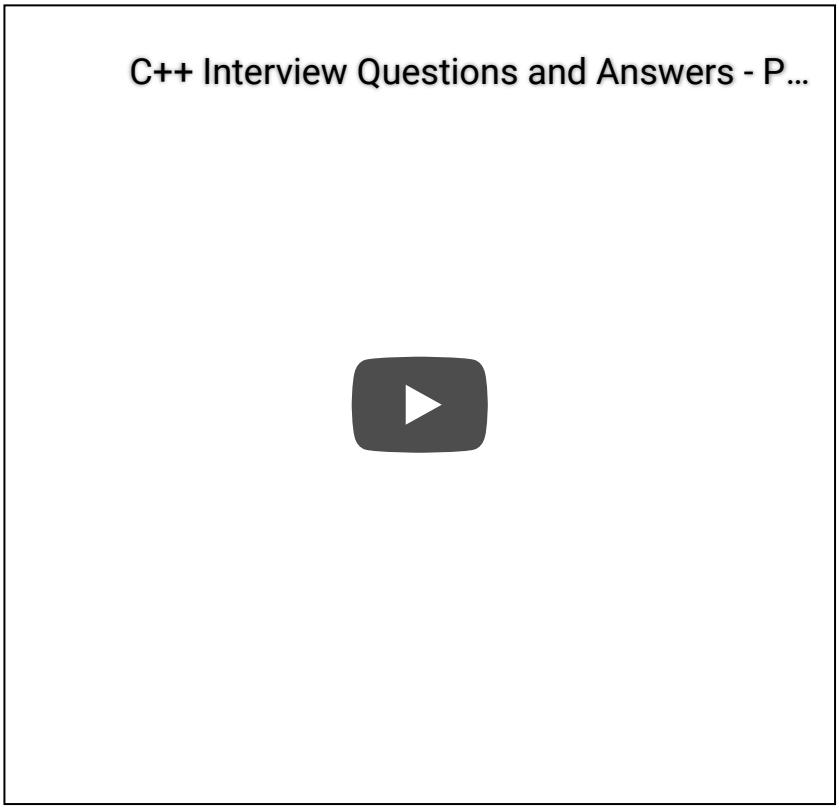
it should be designed from the outside in.

Explain the benefits of proper inheritance.

The biggest benefits of proper inheritance are :
1. Substitutability
2. Extensibility.

1. Substitutability :
The objects of a properly derived class can be easily and safely substituted for an object of its base class.

2. Extensibility :
The properly derived class can be freely and safely used in place of its base class even if the properly derived class is created a lot later than defining the user code.
Extending the functionalities of a system is much easier when you add a properly derived class containing enhanced functionalities.



project.

- Small projects still have a scope to avoid the complete consequence of bad inheritance if the developers communicate and co-ordinate with an easy system design. This kind of a luxury is not possible in big projects, which means that the code breaks in a way difficult and at times impossible way to fix it.

How should runtime errors be handled in C++?

- The runtime errors in C++ can be handled using exceptions.
- This exception handling mechanism in C++ is developed to handle the errors in software made up of independently developed components operating in one process and under synchronous control.
- According to C++, any routine that does not fulfil its promise throws an exception. The caller who knows the way to handle these exceptions can catch it.

When should a function throw an exception?

- A function should throw an exception when it is not able to fulfil its promise.
- As soon as the function detects a problem that prevents it from fulfilling its promise, it should throw an exception.
- If the function is able to handle the problem, recover itself and deliver the promise, then the exception should not be thrown.
- If an event happens very frequently then exception handling is not the best way to deal with it. It requires proper fixation.

Where are setjmp and longjmp used in C++?

properly destruct the local objects.

Are there any special rules about inlining?

Yes, there are a few rules about inlining :
1. Any source files that used the inline function must contain the function's definition.
2. An inline function must be defined everywhere. The easier way to deal with this to define the function once in the class header file and include the definition as required. The harder way is to redefine the function everywhere and learn the one-definition rule.
3. Main() can not be inline.

Explain One-Definition Rule (ODR).

- According to one-definition rule, C++ constructs must be identically defined in every compilation unit they are used in.
- As per ODR, two definitions contained in different source files are called to be identically defined if they token-for-token identical. The tokens should have same meaning in both source files.
- Identically defined doesn't mean character-by-character equivalence. Two definitions can have different whitespace or comments and yet be identical.

What are the advantages of using friend classes?

- Friend classes are useful when a class wants to hide features from users which are needed only by another, tightly coupled class.
- Implementation details can be kept safe by providing friend status to a tightly cohesive class.

What is the use of default constructor?

- When the compiler uses this constructor to create an object – the constructor will have no constructor initializer and a null body.

Differentiate between class and structure.

- The members of structures are public while those of a class are private.
- Classes provide data hiding while structures don't.
- Class bind both data as well as member functions while structures contain only data.

Explain container class.

-Class to hold objects in memory or external storage. It acts as a generic holder.
- It has a predefined behaviour and a known interface.
- It is used to hide the topology used for maintaining the list of objects in memory.
- The container class can be of two types:
1. Heterogeneous container : Here the container class contains a group of mixed objects
2. Homogeneous container : Here the container contains all the same objects.

What is namespace?

- Namespaces are used to group entities like classes, objects and functions under a name.

Explain explicit container.

- These are constructors that cannot take part in an implicit conversion.
- These are conversion constructors declared with explicit keyword.
- Explicit container is reserved explicitly for construction. It

and defines all the valid states for an object.
- When an object is created class invariants must hold.
- It is necessary for them to be preserved under all operations of the class.
- All class invariants are both preconditions as well as post-conditions for all operations or member functions of the class.

Differentiate between late binding and early binding. What are the advantages of early binding?

- Late binding refers to function calls that are not resolved until run time while early binding refers to the events that occur at compile time.
- Late binding occurs through virtual functions while early binding takes place when all the information needed to call a function is known at the time of compiling.
- Early binding increases the efficiency. Some of the examples of early binding are normal function calls, overloaded function calls, and overloaded operators etc.

Explain public, protected, private in C++?

These are three access specifiers in C++.
1. Public : Here the data members and functions are accessible outside the class.
2. Protected : Data members and functions are available to derived classes only.
3. Private : Data members and functions are not accessible outside the class.

Explain Copy Constructor.

It is a constructor which initializes it's object member variable with another object of the same class. If you don't

1. When compiler generates a temporary object.
2. When a function returns an object of that class by value .
3. When the object of that class is passed by value as an argument to a function .
4. When you construct an object based on another object of the same class.

Name the implicit member functions of a class.

1. default constructor
2. copy constructor
3. assignment operator
4. default destructor
5. address operator

Explain storage qualifiers in C++.

1. Const : This variable means that if the memory is initialised once, it should not be altered by a program.
2. Volatile : This variable means that the value in the memory location can be altered even though nothing in the program code modifies the contents.
3. Mutable : This variable means that a particular member of a structure or class can be altered even if a particular structure variable, class or class member function is constant.

Explain dangling pointer.

- When the address of an object is used after its lifetime is over, dangling pointer comes into existence.
- Some examples of such situations are : Returning the addresses of the automatic variables from a function or using the address of the memory block after it is freed.

and reference data members you should use initialization list to initialize them.

When does a class need a virtual destructor?

- If your class has at least one virtual function, you should have a virtual destructor. This allows you to delete a dynamic object through a baller to a base class object. In absence of this, the wrong destructor will be invoked during deletion of the dynamic object.

What is the type of “this” pointer? When does it get created?

- It is a constant pointer type. It gets created when a non-static member function of a class is called.

How would you differentiate between a pre and post increment operators while overloading?

- Mentioning the keyword int as the second parameter in the post increment form of the operator++() helps distinguish between the two forms.

What is a pdb file?

- A program database (PDB) file contains debugging and project state information that allows incremental linking of a Debug configuration of the program. This file is created when you compile a C/C++ program with /ZI or /Zi or a Visual Basic/C#/JScript .NET program with /debug.

You run a shell on UNIX system. How would you tell which shell are you running?

You could also do a ps -f and look for the shell with the highest PID.

What are Stacks? Give an example where they are useful.

A Stack is a linear structure in which insertions and deletions are always made at one end i.e the top - this is termed as Last in First out (LIFO). Stacks are useful when we need to check some syntax errors like missing parentheses.

Differentiate between an external iterator and an internal iterator? What is the advantage of an external iterator.

An external iterator is implemented as a separate class that can be "attach" to the object that has items to step through while an internal iterator is implemented with member functions of the class that has items to step through. With an external iterator many different iterators can be active simultaneously on the same object - this is its basic advantage.

Do you think the following code is fine? If not, what is the problem?

```
T *p = 0;
delete p;
```

- No, the code has a problem. The program will crash in an attempt to delete a null pointer.

In a function declaration, what does extern mean?

- Here, the extern tells the compiler about the existence of a variable or a function, even though the compiler hasn't

functions. How would you do it?

This can be done by using the extern "C" linkage specification around the C function declarations.

Explain STL.

STL stands for Standard Template Library. It is a library of container templates approved by the ANSI committee for inclusion in the standard C++ specification.

What are the different types of STL containers?

Following are the 3 types of STL containers :
1. Adaptive containers : For e.g. queue, stack
2. Associative containers : For e.g. set, map
3. Sequence containers : For e.g. vector, deque

Explain Stack unwinding.

Stack unwinding is a process during exception handling when the destructor is called for all local objects between the place where the exception was thrown and where it is caught.

How would you find out if a linked-list is a cycle or not?

We can find out if the linked-list is not a cycle by using two pointers. One of them goes 2 nodes every time while the second one goes at 1 node each time. If there is a cycle, the one that goes 2 nodes each time will meet the one that goes slower. If this happens, you can say that the linked-list is a cycle else not.

How does code-bloating occur in C++?

- **Free() :** A block of memory previously allocated by the malloc subroutine is freed by free subroutine. Undefined results come out if the Pointer parameter is not a valid pointer. If the Pointer parameter is a null value, no action will take place.

- **Realloc() :** This subroutine changes the size of the block of memory pointed to by the Pointer parameter to the number of bytes specified by the Size parameter and returns a new pointer to the block. The pointer specified by the Pointer parameter must be created with the malloc, calloc or realloc subroutines and should not be deallocated with the free or realloc subroutines. Undefined results show up if the Pointer parameter is not a valid pointer.

Explain Function overloading and Operator overloading.

a. Function overloading :

- The capability of C++ to define several functions of the same name with different sets of parameters is called function overloading. While calling an overloaded function, the C++ compiler selects the proper function by examining the number, types and order of the arguments.
- Function overloading is commonly used to create several functions of the same name that perform similar tasks but on different data types.

b. Operator overloading :

- When the existing C++ operators are redefined to work on objects of user-defined classes, it is called operator overloading.
- Overloaded operators form a pleasant facade which improves the understandability and reduces maintenance

- This is the default storage class.
- Variables in this class are automatically created and initialized when they are defined.
- These variable are then destroyed at the end of the block containing their definition. They are not visible outside that block.

b. Register :
- This is a type of auto variable.
- It gives a suggestion to the compiler to use a CPU register for performance.

c. Static :
- A variable that is known only in the function that contains its definition.
- It is never destroyed and retains its value between calls to that function.

d. Extern :
- This is a static variable.
- Its definition and placement is determined when all object and library modules are combined (linked) to form the executable code file.
- It can be visible outside the file where it is defined.

Explain "const" reference arguments in function?

- It protects you against programming errors that can alter data.
- It allows function to process both const and non-const actual arguments.
- A function without const in the prototype can only accept non constant arguments.
- Using a const reference allows the function to generate and use a temporary variable appropriately.

declaration with a unique namespace that eliminates the potential for those collisions.
- It identifies and assigns a name to a declarative region.
- The identifier in a namespace declaration must be unique in the declarative region in which it is used.
- The identifier is the name of the namespace and is used to reference its members.

Explain Overriding.

- Overriding a method means that replacing a method functionality in child class. To imply overriding functionality we need parent and child classes. In the child class you define the same method signature as one defined in the parent class.
- To override a method, a subclass of the class that originally declared the method must declare a method with the same name, return type (or a subclass of that return type), and same parameter list.
- Method overriding must have same : method name, data type, argument list.

How are virtual functions implemented in C++?

- Virtual functions are implemented using a table of function pointers. It is called the vtable.
- There exists one entry in the table per virtual function in the class.
- The table is created by the constructor of the class.
- When a derived class is constructed, its base class is constructed first. This creates the vtable.
- The derived class may override some of the base classes virtual functions. Such entries in the vtable are overwritten by the derived class constructor. For this

The two pit-falls exist here,
1. If it executes in a member function for an extern, static, or automatic object, the program will probably crash as soon as the delete statement gets executed.
2. When an object finishes like this, the using program might not know about this end. As far as the instantiating program is concerned, the object remains in scope and continues to exist even though the object is finished. Subsequent dereferencing of the pointer can lead to disaster.

Differentiate between a copy constructor and an overloaded assignment operator.

- A copy constructor constructs a new object by using the content of the argument object while an overloaded assignment operator assigns the contents of an existing object to another existing object of the same class.
- Copy Constructor invoke in cases, such as:
i. Creation and initialization of an object simultaneously.
ii. When an object is passed to a function by value.
iii. When an object is returned from a function by value.

Explain Stack & Heap Objects.

The memory a program uses is divided into four areas :
1. The code area :
This is where the compiled program sits in memory.
2. The global area :
The place where global variables are stored.
3. The heap :
The place where dynamically allocated variables are allocated from.
4. The stack :
The place where parameters and local variables are allocated from.

have its own buffers and resources. The destruction of either object will not affect the remaining objects.

b. Shallow copy :
It involves copying the contents of one object into another instance of the same class. This creates a mirror image. The two objects share the same externally contained contents of the other object to be unpredictable.This happens because of the straight copying of references and pointers.

Explain virtual class and friend class.

Virtual Base Class :
- It is used in context of multiple inheritance in C++.
- If you want to derive two classes from a class, and further derive one class from the two classes in the second level, you need to declare the uppermost base class as 'virtual' in the inherited classes.
- This prevents multiple copies of the uppermost base class data members when an object of the class at the third level of hierarchy is created.

b.) Friend class:
- When a class declares another class as its friend, it is giving complete access to all its data and methods including private and protected data and methods to the friend class member methods.
- Friendship is not necessarily bi-directional. If A declares B as its friend it does not imply that A can access private data of B. It only means that B can access all data of A.

Explain the scope of resolution operator.

- A scope resolution operator (::) is used to define the member functions of a class outside the class.

List the advantages of inheritance.

- Inheritence permits code reusability.
- Reusability saves time in program development.
- It encourages the reuse of proven and debugged high-quality software which reduces the problems after a system becomes functional.

Differentiate between declaration and definition.

- The declaration informs the compiler that at some later point we plan to present the definition of this declaration.

What do you mean by a template?

- Templates enable creation of generic functions that admit any data type as parameters and return value without having to overload the function with all the possible data types unless they fulfill the functionality of a macro.
- Its prototype can be one of the following :

```
template function_declaration; template
function_declaration;
```

- The only difference between both prototypes is the use of keyword class or typename. It's use is indistinct since both expressions have exactly the same meaning and behave exactly the same way.

Explain RTTI.

- RTTI - Runtime type identification
- It lets you find the dynamic type of an object when you have only a pointer or a reference to the base type.
- RTTI is the official way in standard C++ to discover the type of an object and to convert the type of a pointer or reference.

For Example : Integer i2(i1);

- An assignment operator doesnot invoke the copy constructor. It simply assigns the values of an object to another, member by member.

Difference between struct and class in terms of Access Modifier.

Classes and structures are syntactically similar. In C++, the role of the structure was expanded, making it an alternative way to specify a class. In C, the structures include data members, in C++ they are expanded to have function members as well. This makes structures in C++ and classes to be virtually same. The only difference between a C++ struct and a class is that, by default all the struct members are public while by default class members are private.

What are virtual functions?

Polymorphism is also achieved in C++ using virtual functions. If a function with same name exists in base as well as parent class, then the pointer to the base class would call the functions associated only with the base class. However, if the function is made virtual and the base pointer is initialized with the address of the derived class, then the function in the child class would be called.

What is a stack? How it can be implemented?

A Stack is a linear data structure which is a collection of homogenous elements but where insertion and deletion operations take place at one end only, called TOP of the stack. Stack is also known as LIFO (Last In First Out) structure as the element which is inserted in the last will be the first element to come out from the stack.

A modifier is also known as mutator where the change in the member function value will change the value of a data member that is associated with the function. It modifies the state of an object that is why mutator as it mutates or makes duplicate copies of itself to be used so that by one function many things can be changed at a time.

What are the various operations performed on stack?

Various operations which can be performed on stack are as follows :
creatempty() : It creates an empty stack by initializing TOP to -1.
Isempty() : It determines whether stack is empty or not. It returns value 1 if stack is empty otherwise return 0.
Push() : Adding a new element at the top of the stack is called Push.
Pop() : Removing an element from the top of the stack is called Pop.

How a new element can be added or pushed in a stack?

Pushing an element into stack :

Whenever an element is to be pushed into stack, TOP is increased by 1 and then the element is inserted in the linear array denoted by S[ ] at the location with index TOP i.e. at S[TOP]. During push() operation, a stage may come when TOP points to the last location in array S[ ] i.e. it becomes equal to stksize-1, then no more element can be pushed into stack and we say that Stack Overflow.

```
Algo Push(S,Top, Item)
Step 1 : If (Top == stksize-1)
then (i) Write "Stack Overflow "
```

Explain Queue. How it can be implemented?

- A queue is a linear data structure which is a collection of homogeneous elements where insertion and deletion occurs at different ends.
- The end where insertion takes place is called REAR and the end where deletion takes place is called FRONT.
- Queue is also known as FIFO (FIRST IN FIRST OUT) because the element which is inserted first last will be the first element to come out from the queue.
- Queue can be implemented by :
1. Array Implementation of Stack
2. Linked List Implementation of Stack

What are the four partitions in which C++ compiler divides the RAM?

1. Stack Area :
This part of memory is used to store formal parameters, local variables, return addresses of function call etc.
2. Program Code Area :
This part of memory is used to store the object code of the program.
3. Global Variable Section :
This part of memory is used to store global variables defined in the program.
4. Heap Area or Free Storage Pool :
It consists of unallocated memory locations which are allocated dynamically during program execution using new operator.

Explain static and dynamic memory allocation with an example each.

- When amount of memory to be allocated is known beforehand i.e. at the time of compilation, it is known as

beforehand, rather it is determined at the time of program run, it is called Dynamic Memory Allocation. It leads to efficient utilization of storage space.

- **Example :**

```
cout << " Enter number of elements: ";
cin >> N;
int *A = new int[N]; // dynamic memory allocation
```

What is function prototyping? What are its advantages?

- Function prototyping is a function declaration statement that tells the compiler about the return type of the function and the number as well as type of arguments required by the function at the time of calling it.

- **Syntax :**

```
return_type function_name( type1 arg1, type 2 arg2, ... );
```

Advantages of function prototype :

- It helps the compiler in determining whether a function is called correctly or not. Each time when a function is called, its calling statement is compared with its prototype. In case of any mismatch, compiler reports an error.
- A function must be defined before calling it. But prototyping allows a function to be called before defining it.

Write a program using Display() function which takes two arguments.

```
cout << A[0] << " ";
cout << A[1] << " ";
.....
cout << A[N-1] << " ";

In general : cout << A[i] << " " ; where i = 0 to N-1

void Display( float A[ ], int N )
```

Write a program using SHIFT_HALF( ) function to shift the elements of first half array to second half and vice versa.

```
swap(A[0], A[mid])
swap(A[1], A[mid+1])
....
swap(A[mid-1], A[N-1])

mid = n/2; i = 0, j = mid; swap( A[i], A[j] ) where i = 0 to mid-1

void SHIFT_HALF( float A[], int n )
{
   int mid= n/2;
   for ( int i = 0, j =mid; i<= mid-1; i++,j++)
   {
      float T = A[i];
      A[i] = A[j];
      A[j] = T;
   }
}
```

What is searching? Explain linear and binary search.

- Finding the location of a given element in a given data structure is called searching.

- There are two types of search :

1. Linear Search :
In this, the element to be searched is compared one by one with each element of given list, starting with first element. The process of comparisons remain continue until the element is not found or list gets exhausted.

ascending or descending order.
ii. The list must be of finite size and should be in form of linear array.

Explain Selection sorting. Also write an example.

- In selection Sorting, one has to perform N-1 iterations or steps to sort a linear array containing N elements.
- In first iteration, we select the first minimum value and interchange it with the element present at first position.
- In Second iteration, we select the Second minimum value and interchange it with the element present at second position and so on.
- **For example :**

```
void SSORT( float A[ ], int n )
{
  for ( int i = 0; i<= N-2; i++)
  {
     int min = i;
     for ( int j = i+1; j<= N-1; j++)
     if ( A[min] > A[j] )
        min = j;
     // end of j loop
     if ( i != min )
     {
        float temp = A[i];
        A[i] = A[min];
        A[min] = temp;
     }
  }
}
```

Explain Bubble sorting.

In Bubble Sorting, we have to perform N-1 steps to sort a linear array.
1. In first iteration, we compare A[0] with A[1], A[1] with

A[N-3] with A[N-2] and interchange them if they are not in desired order.
3. In Third iteration, we compare A[0] with A[1], A[1] with A[2],.....
A[N-4] with A[N-3]and interchange them if they are not in desired order.
4. In last iteration , we compare A[0] with A[1].

What is Insertion sorting?

- In insertion sorting, an array is divided into two parts :
1. Sorted part
2. Unsorted part
- Initially, sorted part contains only one element i.e. A[0] and unsorted part contains remaining N-1 elements i.e. A[1], A[2], ... A[N-1]. We pick elements form unsorted part one by one and insert them in the sorted part. Thus, we have to perform N-1 iterations to sort a linear array.

Write a program using MERGE () function to combine the elements of array X[ ] and Y[ ] into array Z[ ].

- Array X[ ] contain M elements and array Y[ ] contains N elements.

```
void MERGE( int x[], int y[], int z[], int m,int n )
{
  int L = 0, R = m+n-1;
  for ( int i = 0; i<= m-1 ; i++)
  {
     if ( x[i] % 2 == 0 )
     {
        c[R] = x[i];
        R--;
     }
     else
     {
        C[L] = x[i];
```

```
        {
          c[R] = y[i];
          R--;
        }
        else
        {
          C[L] = y[i];
          L++;
        }
    }
}
```

What are the various situations where a copy constructor is invoked?

Various situations where a copy constructor is invoked are as follows :
- When an object is defined and initializes with the values of another object of the same type, then copy constructor is invoked.
- When an object is passed by value method, then copy constructor is invoked to create the copy of the passed object for the function.
- When a function returns an object, then copy constructor is invoked to create a temporary object to hold the return value in the memory.

What is the need of a destructor? Explain with the help of an example.

- During construction of an object, resources may be allocated for use.
- For example, a constructor may have opened a file and memory area may be allocated to it. These allocated resources must be deallocated before the object is destroyed.
- A destructor performs all clean-up tasks like closing a

```
    int i, j;
    public :
    X( int m = 5, int n = 10)
    {
        i = m;
        j = n;
    }
    ~X() // destructor
    {}
    void print()
    {
        cout << i << " " <, j << endl;
    }
};
```

List the special characteristics of constructor.

- A constructor has the same name as that of class.
- It is automatically invoked when an object of the class is declared.
- Constructor obeys the usual access rule. Private & protected constructor can only be accessed by the member function and friend function of the class, Public constructor is available for all the function. Only that function can create the object that has access to the constructor.
- No return type is specified for the constructor.
- These cannot be inherited but a derived class can invoke base class constructor.
- A constructor can also have default arguments.
- A constructor can invoke the member functions.
- The default constructor and copy constructor are provided by the compiler only if these are not defined by the programmer.

1. Implicit Calling :

By implicit calling, we mean that the constructor's name is not specified in the calling statement.

General Form :

```
class_name object_name( value1, value2, .... );
```

Example :

```
X o1(4,5);
```

2. Explicit Calling :

By explicit calling, we mean that the constructor's name is specified in the calling statement.

General Form :

```
class_name object_name = constructor_name( value1, value2, .... );
```

Example :

```
X o1 = X(4,5);
```

What are the rules for naming an identifier?

- An identifier is the user defined name given to different elements in a program via: - variable name, class name, function name, array name etc.
- Example: M1, M2, M3, Rad. Ht, Vol, A, Calculate
- Identifiers are infinite in numbers.
- An identifier can be named in following ways :

1. An identifier's name must begin with an alphabet or underscore(_).
2. It should not contain special symbol except underscore.
3. It should not be a keyword.

What are the different types of comments allowed in C++?

A line of text beginning with symbol double slash (//) up to the end of line is called single line comment. Comments are ignored by compiler during compilation.

2. Multiple line comments :

The text which is enclosed between a pair of symbols (/* and */) is called Multiple line comments.

What is the difference between a copy constructor and an overloaded assignment operator?

A copy constructor is used within the class to copy one object in a new object. This can be done by using the argument object in it, whereas an overloaded assignment operator link an existing object to another object which is existing but in the same class.

How to implement is-A and has-A class relationships?

Is-a is a class relationship which is specialization of another class. It is used to describe the relationship with other classes. This relationship can be seen implemented in inheritance.

For example if an employee class is there with a person then the employee is-A person.

Has-A shows the relationship between classes. In this, class can contain instances of another class. **For example**, employee "has" income, so the employee class is having has-A relationship with the Salary class.

Why use of template is better than a base class?

when the type of the class is unknown to the user of manager of the class.

What are the advantages of using a pointer? Define the operators that can be used with a pointer.

Advantages of pointer :

- Through pointer, one can access a memory location directly and manipulate it.
- Pointers support dynamic memory allocation using new & delete operator.
- A pointer makes execution of certain routines faster.
- The operators that can be used with a pointer:

At Address operator (*) :

This operator gives the rvalue of the memory location whose address is stored in its operand.

For example: If P is a pointer that hold the address of variable A, then *P gives the rvalue of the variable A.

ADDRESS operator (&) :

This operator gives the address of its operand.

For Example: &A gives the address of variable A.

Show the application of a dynamic array with the help of an example.

- Dynamic array is an array whose size is defined during the program execution.
- Declare a dynamic array :

```
data_type *array_name = new data_type[size];
```

Example :

```
int N;
cout << " Enter the number of elements: ";
cin >> N;
```

```
  cout << " Enter the number of elements : ";
  cin >> N;
  int *A = new int[N];
  for (int i = 0; i < N; i++)
  {
     cout << “Enter the element no. " << i+1 << " : ";
     cin >> *(A+i);
  }
  clrscr();
  cout << " Given list is as follows: \n\n";
  for ( i = 0; i < N; i++)
  cout << A[i] << " ";
  getch();
}
```

Define linked lists with the help of an example.

- It is a linear data structure which is a collection of homogeneous elements called Nodes where linear relationship is maintained by using Pointers.
- **Example :**

```
Declare a linked list containing integer values.
struct Node
{
  int data;
  Node *link;
};
class Linked
{
  Node *start;
  public:
  Linked()
  {
     start = NULL;
  }
  void Insert();
  void Delete();
```

Explain the virtual inheritance in C++?

Virtual inheritance is used when a single base class is inherited with virtual methods. It can be achieved by the virtual keyword in the program. In this, the object that belongs to virtual class becomes common to the base class. It is used for multiple inheritance, as it creates multiple sub objects and gives the feature where a class can inherit from more than one classes.

Why is Standard Template Library used?

The Standard template library is used as a container to the templates which have been approved by the ANSI. It includes the standard C++ specification. It helps to construct programming in object oriented manner. It allows the use of pre-defined libraries for generic programming model. It allows faster execution of the programs and allow the user to use functions without even writing them.

What problem does the namespace feature solve?

Namespace is an identifier that provides multiple libraries. This is used to remove the name collision when a name is linked with two or more libraries. It includes the external declarations of the library with unique namespaces so it eliminates the potential of the collision.

Next Page »

When is dynamic checking necessary? - C++

When is dynamic checking necessary? - Whenever the definition of a variable is not necessary before its usage....

Define structured programming - C++

Structured programming - Structured programming techniques use functions or subroutines to organize...

Explain object oriented programming.

Object oriented programming - Object oriented programming uses objects to design applications...

Post your comment

Discussion Board

Good

Very useful questions of C++.
But page is not so attractive. Please, change the UI
Pratik 08-15-2015

#14 is flat out wrong

#14 says:
"No, the code has a problem. The program will crash in an attempt to delete a null pointer. "

Contrary to popular believe, this is flat out incorrect. Calling delete on a null pointer is harmless. If you don't believe me, go write any of the following code and run it:

```
delete NULL;

delete nullptr;

int* foo = NULL;
delete foo;
```

I would recommend removing #14 from the list of questions.
John Selbie 06-12-2014

for example:

```
class B
{
double value;
public:
B(int i )
operator double()
{
return value;
}
};

B BObject;

double i = BObject; // the conversion operator is called to assign the value.
```

What is diff between malloc()/free() and new/delete?

• The malloc allocates memory for object in the heap but not invokes object's constructor for initiallizing the object.

• new also allocates thememory and also invokes constructor to initialize the object.

• malloc() and free() are not able to support object semantics but does not construct and destruct any objects

• string * ptr = (string *)(malloc (sizeof(string)))

• int *p = (int *) (malloc (sizeof(int)));

• int * p = new int;

new and delete,both can be overloaded in a class :

• "delete" first calls the object's termination routine and then releases the space the object occupied on the heap memory.

delete []my_ints;

Define macro.

• There is no way for the compiling to verify that the macro parameters are of compatible types.

• The macro can be expanded without any special type checking.

• If macro parameter is having a post incremented variable (like c++), the increment will be performed twice.

for example:

Macro:

#define min(i, j) (i < j ? i : j)

template:
template
T min (T i, T j)
{
return i < j ? i : j ;
}

What are C++ storage classes?

storage classes types are :

• auto: This the default . Variables are naturally created and initialized when they are defined and are destroyed at the end of the block contained their definition.

• register: It's a type of auto variable. This helps the compiler to use a CPU register for performance

• static: It's a variable that is known only in the function that contains its definition but is never destroyed and retains its value between calls to that function.

• If the derived class overrides the base class method by redefining the same function, then if client will want to access redefined the method from derived class through a pointer from base class object, then we have to define this function in the base class as a virtual function.

```
class parent
{
void Show()
{
cout << "i'm parent" << endl;
}
};

class child : public parent
{
void Show()
{
cout << "i'm child" << endl;
}

};

parent * parent_object_ptr = new child;

parent_ object_ ptr -> show() // calls parent->show() i

now we goto virtual world...

class parent
{
virtual void Show()
{
cout << "i'm parent" << endl;
}
};

class child : public parent
{
void Show()
{
cout << "i'm child" << endl;
```

parent_object_ptr->show() // calls child->show()

What is pure virtual function? or what is abstract class?

While we define function prototype in a base class without implementation .The base class is called abstract class

Example of a pure virtual function or abstract class this way..

```
class B
{
void f() = 0;
}
```

B MyB; // compilation error
Advanced C++ interview 07-18-2012

Basic C++ interview questions - Frequently asked C++ interview

Write some differences between an external iterator and an internal iterator? Describe the advantage of an external iterator.

• An external iterator gets implemented as a separate class that can be "attach" to the object which is having items to step through .

• In case of an internal iterator it is implemented with a member function of the class which are having the items to step through.

• With the help of an external iterator many different iterators can be activated simultaneously on the same object.

Verify the following code. Point out the problems.

T *p = 0;

delete p;

• No, the code has a problem.

• The program will be crashed for an attempt to delete a null

of a function.

• The variable or function can be defined within another file

How can you link a C++ program to C functions?

This can be done Intwo methods ;

• First by using the extern "C" linkage specification

• The linkage is done in the C function declarations.

Define STL.

• STL stands for Standard Template Library.

• It is the library for container templates.

• This is approved by the ANSI committee for including in the standard C++ specification.

Name the different types of STL containers.

The 3 types of STL containers are:

• Adaptive containers - e.g. stack, queue,

• Associative containers - for e.g. set, map

• Sequence containers - e.g. vector, deque

What is Stack unwinding?

• Stack unwinding is the process for exception handling

• It takes place when the destructor is being called.

• The destructor calls all the local objects between the place where the exception had been thrown and where it had been caught.

How come you find out if a linked-list is a cycle or not?

• Incase there is a cycle, the one that goes 2 nodes each time will meet the one that goes slower. If this occurs, we can confirm that the linked-list is a cycle else not.

Define a nested class. Explain how it can be useful.

• A nested class is said to be a class which is enclosed in the scope of another class.

For example:

```
// Example : Nested class
//
class OuterClass
{
class NestedClass
{
// ...
};
// ...
};
```

• Nested classes are of great use for organizing code and for controlling access and dependencies.

• Nested classes do obey access rules.

So, if NestedClass is when public ,any code can be named as OuterClass::NestedClass.

When does the C++ compiler create temporary variables?

If the function parameter is a "const reference", the compiler generates temporary variables in the following 2 ways.

• a) If the actual argument is the correct type, but it isn't Lvalue

```
double Cubes(const double & num)
{
numb = numb * numb * numb;
```

double value = cubes(3.0 + temp); // the argument is said to be an expression ,not the Lvalue;

• b) A type can be converted to the correct type

long temp = 3 L;

double value = cuberoot (temp) ;

Explain the differences between List x; & List x();.

There exists a big difference which is explained via a code below:

• Let, List is a name of any class.

Then function f() evokes a local List object x:

void f()

{

List x; // Local object x

...

}

But the function g() invokes f() which eventually returns a List:

void g()

{

List x(); // function which returns the List

...

}

```
class B
{
public:
B( int i );
};
```

B B Object = 10 ; // assigning int 10 B object

Basic C++ interview 07-18-2012

C++ interview questions and answers for freshers

Define Copy Constructor.

• It is a simple constructor.

• With a different object of the same class, it initializes it's object member variable.

• If we don't implement copy constructor, it does automatically.

When do we use copy constructors?

Copy constructors are called in these scenarios below:

• At the time of generating a temporary object by the compiler.

• If a function is returning an object of that particular class by value

• At the time of passing by value as an argument to a function by the object of that class

• At the time of constructing an object based on a different object of the same class

What are the implicit member functions of class?

The implicit member functions are

• default ctor

It is mainly used for initializing.

• copy ctor

• default destructor

It acts just like the inverse of constructor for de-initializing variables.

• address operator

It connects an operand to its pointer variable.

What are the storage qualifiers?

The storage qualifiers are:

• Const - If the memory gets initialized once, it will remain indifferent.

• Volatile - If the value in the memory location will be altered though nothing is changed in the program code, this value may be changed.

• Mutable - This means that if a member of a structure or class can be altered though a particular structure variable, class, or class member function will remain constant.

What is dangling pointer?

• Pointer that does not point to a proper or valid object of the correct type is called Dangling pointer.

• If the address of an object is used after its lifetime, the concept of dangling pointer will come.

The examples of such situations can be given as:

• To return the addresses of any automatic variable from the function or by using the address of the memory block after it has got freed.

Do we have to use initialization list in spite of the assignment in constructors?

• We can use non-static const data members and reference data

If our class has at least one virtual function, we have to have a virtual destructor.

• This will allow deleting a dynamic object.

• In absence of this, the wrong destructor will be invoked during deletion of the dynamic object.

What is type of “this” pointer? Explain when it is get created?

• "this' is a constant pointer type.

• It will get created if a non-static member function of the class is called up.

How we can differentiate between a pre and post increment operators during overloading?

• We have to mention the keyword int as the second parameter

• It is to be mentioned the post increment form of the operator++() .

Define a pdb file.

• It's a program database (PDB) file.

• This file contains debugging and project state information which does the incremental linking of a Debug configuration for the program.

• This file gets created at the time of compiling a C/C++ program with the help of /ZI or /Zi or a Visual Basic/C#/JScript .NET program with /debug.

When do we run a shell in the UNIX system? How will you tell which shell you are running?

• For checking this we can simply run the command Echo $RANDOM.

Thus the results obtained will be:

• Undefined variable if we'll do this in the C-Shell.

• We can also run a ps -l and look for the shell having the highest PID.

Define Stacks. Provide an example where they are useful.

• A Stack is of course a linear structure.

• In stack insertions and deletions are being made at the one end i.e the top - which is termed as last in, first out (LIFO).

• Application of Stacks can be used when we need to debug some syntax errors like missing parentheses.
Freshers C++ questions 07-18-2012

C++

Answer of question 14 is wrong.
Code will work fine. There should be no problem in deletion of NULL pointer.
If pointer is uninitialized then only it will crash.
Madhumita 03-20-2012

C++ interview questions and answers

What is the type of "this" pointer?

It is a constant pointer.

When does a "this" pointer get created?

When a non-static member function of a class is called.

Define VPTR.

The address of the VTABLE stored in the object is known as VPTR.

Define upcasting.

Storing the address of the derived class object in the base class pointer.

Explain how to initialize a const data member.

The const data member can be initialized with constructor initializer list.

Java has garbage collection whereas C++ does not.

Define virtual constructor.

There is no such concept in C++.

Define anonymous class.

A class that is defined without any name is called anonymous class.

Explain how to initialize a const member data.

Use constructor initializer list.

Difference between delete and free.

Delete invokes destructor, free will not
Kavita 12-5-2011

Related Content

- C++ - Part 1
- C++ - Part 2
- C++ Constructors and Destructors
- C++ Containers
- C++ Derived Class
- C++ Data Types
- C++ Control Constructs
- C++ Collections
- C++ Functions
- C++ Arrays
- C++ and C String
- C++ Classes Structure
- C++ Friend Functions Classes
- C++ Polymorphism
- C++ Multiple Inheritance
- C++ Function Template