Interview Questions

Contents

Contents

[TABLE-OF-CONTENTS]

1 Design and Architecture.

1.1 Object Oriented Design and Architecture

1.1.1 Access Modifiers

1. Is private member of an instance accessible with other class with same type ?

```
using System;
class cls {
    private int i;
    public int I {
get { return i;  }
set { i = value; }
    }
    public cls() {}
    public cls (cls right) {
i = right.i;        // No Error: Private member of same type is accessible
    }
}
class program {
    public static void Main () {
cls a = new cls ();
a.I = 10;
cls b = new cls (a);
Console.WriteLine (b.I);
    }
}
```

Code Reference: Access Modifier

1.1.2 Static

1. Can static class in C#.NET can inherit from other class ? Can Static class be inherited by other classes ? ANS)

   - Contains only static members.
   - Cannot be instantiated.
   - Is sealed.
   - Cannot contain Instance Constructors.
   - Static classes are sealed and therefore cannot be inherited. They cannot inherit from any class except Object. Static classes cannot contain an instance constructor; however, they can contain a static constructor. Non-static classes should also define a static constructor if the class contains static members that require non-trivial initialization.
   - `http://msdn.microsoft.com/en-us/library/79b3xss3.aspx`

2. Singleton vs Static class . Singleton VS Static class

1.1.3 Lazy Initialization

1. What is Lazy Initialization ?

2. What is Double-Checked-Locking ?

   (a) Why volatile is used with Double-Checked-Locking ? Double-Checked$_{\text{Initialization}}$-Wikipedia-Online Double Checked Initialization Wikipedia offline

   (b) What is difference between Lazy and LazyInitializer class ? Lazy Initialization Albahari `file:///C:/Users/Pralay/Google%20Drive/SoftwareDevelopment/Books_And_Tutorials/DotNet/books/Threading/Threading_by_Albahari/www.albahari.com/threading/part3.html#_Lazy_Initialization`

1.1.4 Inheritance

1. What are the advantages and disadvantages of multiple inheritance ?

   (a) How can we achive this via delegation ?

1.2 Design Pattern

1.2.1 Explain Visitor design pattern.

1.2.2 IOC.

1. What are the benefits of IOC ?
2. What is the difference between IOC and DI ?
3. How IOC simplify unit testing ?

1.2.3 What is Observer pattern. How we can implement it in c# if we don't have events ?

1. What is difference between observer pattern and publisher subscriber pattern?

1.2.4 What is MVVM ?

MVVM Pattern

1.2.5 What is MVC ?

MVC Pattern

1.2.6 What is worker thread pattern ? # TODO

Worker Thread Pattern (Chapter from Book)

1.3 Singleton Pattern

1.3.1 2

1.3.2 1

1.3.3 2

1.4 Design a website which provides pizza . It provides daily offer. How to implement Price class when there are dynamic Offers ?

1.5 Write a code to represent Address Book with name,contact number and email. IT can be search able by all the members.

1.6 Software Development Metodologies.

1.6.1 What are different types of Software Development Methodologies ? What is Agile Methodology ? What is TDD ? What is Scrum ?

- References: Agile and other methodologies

1.6.2 What is continuous integration ?

- Reference: Continuous Integration

1.6.3 What is TestDrivenDevelopment ?

- Reference: Test Driven Development

---

2 C++ Questions

2.1 Access Modifier

Stackoverflow article

2.1.1 What is the use of private destructor ?

Ans) The class can't be instantiated.

2.1.2 What is the use of protected destructor ?

Ans) The class can't be instantiated only derived class can be instantiated.

2.2 Threading

2.2.1 Timer in C++.

1. What are the different timer classes provided in boost and standard c++ ?

2. Does it has a separate timer thread as csharp ? How does it works ?

    - Code Reference: Timers

2.2.2 Synchronization

1. Lock Free Algorithm

    (a) What is lock free algorithm ? How can we implement it ?

    (b) Implement a thread safe lock free queue.

    - Reference: Lock-Free
    - Code Samples: Lock-Free

2. What are different types of synchronization objects provided by windows and Linux ?

    - Reference:Windows and Linux Synchronization Objects

3. Deadlock

4. What is a deadlock ? Explain the scenario(s) where you faced deadlock . How did you resolved it ?

    - Reference: Deadlock

2.2.3 Threading vs Parallel programming.

2.3 Smart Pointers

2.3.1 What are different types of smart pointers ?

- Reference Smart Pointers in C++

2.4 Assignment Operator

2.4.1 What is difference between copy constructor and assignment operator ?

2.4.2 What are the precautions developer need to to take while implementing assignment operator ?

2.4.3 What is the signature of assignment operator ?

1. What assignment operator returns ? Does it return reference or Pointer and Why ?

2. Does it take pointer or reference as parameter ? Why ?

2.4.4 Implement assignment operator for MyString class.

- Reference: MyString class

2.5 Bind

2.5.1 What is the use of bind ?

2.5.2 When we have functors why we have bind ?

2.6 Constant and Internal linkage.

2.6.1 What is internal linkage ?

2.6.2 What is mutable and immutable objects in cpp ?

2.6.3 Are const static in c++ ?

2.6.4 syntax and meaning for *constant pointer* and *pointer to constant* ?

2.6.5 What is mutable keyword in cpp ? How and when it is used ?

2.6.6 How to remove constness using casting in cPP ?

- Reference: Constant and Internal Linkage

- Code Reference: const

2.7 Functors

2.7.1 What are functors or function objects ?

2.7.2 What are the benefits compared to function pointers ?

2.7.3 Bind vs Functors ?

2.8 Lambda

2.8.1 What is the syntax of lambda in cpp ?

2.8.2 What square bracket '[]' means in lambda ?

2.8.3 Does lambda in c++ has closure property ?

- Code Reference: lambda

2.9 asyn

2.9.1 What is async ?

Ans: async is a asynchronous callable objects. It runs as background task.
Code Sample: async

2.10 Templates

2.10.1 What is specialization and partial specialization ? Provide real time sample of both .

- Code Samples: templates

---

3 .Net Questions

3.1 AppDomain

3.1.1 What is AppDomain ?

3.1.2 AppDomain and Thread ?

3.1.3 AppDomain and Security ?

3.1.4 Inter-AppDomain Communication ? InterProcess Communication and Appdomain ? Remoting ?

- AppDomain Code Samples

3.2 Why constant are static?

3.3 Generics

3.3.1 What is the differenece between Covariance and Contra-variance ?

- Given a Sample :
  - We have Employee class and Manager class .
  - Manager is derived from Employee.
  - We have a method which takes a Employee list as parameter.
  - Can List of Manager be passed to the above method ?
  - What if we are dealing with array i.e. in the above methods we use array and not List ?

Ans) - References: Covariance and Contra-variance

3.4 Object class

3.4.1 What are the different methods in object class?

1. What are different non-virtual, protected, virtual and static methods ?

3.4.2 Is everything derived from object ?

1. What about primitive types like int long etc ?
   - Ans) Yes even primitive types.

3.5 String

3.5.1 Describe efficient string builder implementation.

3.6 Virtual Methods

3.6.1 If virtual method called in constructor ?

- In C# virtual method of derived will be called.
- It s is not safe as Derived class constructor is not called yet and this derived call is not properly initialized.

3.7 static vs non-static method ?

3.7.1 What are problems while unit testing static methods/classes ?

3.8 Can a constructor be static ?

http://msdn.microsoft.com/en-us/library/k9x6w0hc.aspx

3.9 Mutable vs Immutable objects.

3.9.1 What are immutable objects in .Net ?

3.9.2 What are the benefits of immutability ?

3.10 Object Identity and Equality

3.10.1 What is difference between object equality and object identity ?

1. If a = b that a.hash = b.hash and visa versa ?
2. How to override equal functionality ?
3. When we have to override GetHashCode ? Why ? Provide a sample .
4. By default does Equals method check reference ? If yes than what is the use of ReferenceEqual ?
5. Why operator== is not defined in object class ? Reference to 'Effective C#' chapters

3.11 How to create a deep copy og object in c# ? What is ICloneable interface and its pitfall ?

Reference to chapter from 'Effective C#'

3.12 Threading

3.12.1 What are uses of threading ? In Which scenario a thread need to be spawned ?

1. Maintaining a responsive user interface By running time-consuming tasks on a parallel worker thread, the main UI thread is free to continue processing keyboard and mouse events.

2. Making efficient use of an otherwise blocked CPU Multithreading is useful when a thread is awaiting a response from another computer or piece of hardware. While one thread is blocked while performing the task, other threads can take advantage of the otherwise unburdened computer.

3. Parallel programming Code that performs intensive calculations can execute faster on multicore or multiprocessor computers if the workload is shared among multiple threads in a divide-and-conquer strategy (see Part 5).

4. Speculative execution On multicore machines, you can sometimes improve performance by predicting something that might need to be done, and then doing it ahead of time. LINQPad uses this technique to speed up the creation of new queries. A variation is to run a number of different algorithms in parallel that all solve the same task. Whichever one finishes first wins this is effective when you cant know ahead of time which algorithm will execute fastest.

5. Allowing requests to be processed simultaneously On a server, client requests can arrive concurrently and so need to be handled in parallel (the .NET Framework creates threads for this automatically if you use ASP.NET, WCF, Web Services, or Remoting). This can also be useful on a client (e.g., handling peer-to-peer networking or even multiple requests from the user).

`file:///C:/Users/Pralay/Google%20Drive/SoftwareDevelopment/Books_And_Tutorials/DotNet/books/Threading/Threading_by_Albahari/www.albahari.com/threading/index.html#_Threadings_Uses_and_Misuses`

3.12.2 What preempted means ?

3.12.3 What is Reentrancy ?

3.12.4 Static method and Thread .

3.12.5 What is the difference between Thread.Sleep(0) and Thread.Yield ?

1. What are the uses of Thread.Yield ?

3.12.6 TPL

- Task Summary
- Code Samples

1. Task
   (a) What is task ?
2. Task Factory/Group Tasks/Parent Child relationship.
3. TaskScheduler
4. ContinueWith
5. Delay
6. CancellationTokenSource/ConcellationContext.
   (a) How can we cancel a task ?
7. Handle exceptions in thread.
8. ExecutionContext
9. Parallel ForEach

3.12.7 Timer

- Code Reference: Timer

1. What is delay?
2. What is problem with the Threading.Timer which is resolved by delay ?

3.12.8 Threading vs Parallel programming.

3.12.9 What is Thread Affinity ?

3.12.10 What is worker thread ?

Ans) Wroker thread is a pattern. It has two variants.

1. Single thread listeneing to queue of task. Task are interted by client.
2. Thread Pools listeneing to queue of task.

`Worker Thread Pattern (Chapter from Book)`

3.12.11 What is the difference between UI and worker thread ?

3.12.12 What is difference between foreground and background thread?

1. What will happen if in the main method we spawn a background thread and without waiting main terminates ?
2. What will happen if in the main method we spawn a foreground thread and without waiting main terminates ?

Foreground Thread	Background Thread
Thread will run even after main goes out of control.	The thread will terminated once main go
The application will not terminate.	The application will terminate and not w
	There will be no exception even if it term
	Finally will not be called so resource (un

3.12.13 What is backgroung worker thread ?

Code Sample: Background VS Foreground Threads

3.12.14 When thread is deemed block ?

1. What is the difference between *Blocking* and *Spinning* ?

   Blocking and Spinning

3.12.15 Abort and interrupt:

1. What abort and interrupt does ?
2. Does exception occurs in the thread when abort or interrupt is called ? If yes what are the name of the exception for abort and interrupt resp ?
3. Is the abort or interrupt exception once handled thrown agian ? if yes why ?
4. What happen if the thread is in suspended mode and abort or interrupt is called on the thread ?

Abort
It is used to release a blocked thread prematurely; for instance, when ending an application.
The Abort method is also capable of ending a nonblocked thread.
A blocked thread can also be forcibly released via its Abort method. This has an effect similar
Furthermore, the exception will be rethrown at the end of the catch block (in an attempt to ter
There are two cases, though, where you can safely use Abort. One is if you are willing to tear d
The big difference between Interrupt and Abort is what happens when it is called on a thread t
ThreadAbortException even unhandled never propogates to AppDomain.UnhandleExceptionEv

`file://c:/P:/Books_And_Tutorials/DotNet/books/Threading/Threading_by_Albahari/www.albahari.com/threading/part3.html` and Abort

Interrupt and Abort Introduction Abort Thread details

3.12.16 Synchronization

1. Blocking Methods
   (a) Wait
   (b) Join
   (c) Sleep
   (d) What is IO bound asynchronous operation ?
      i. What is async and await ?
         - Reference: IO-Bound-Asynchronous-Operations
         - Code Reference: IO-Bound-Async-Operations
2. Locking constructs
   (a) Semaphore
   (b) Simple Hybrid Lock
   (c) What is SemphoreSlim ?
   (d) What is ReaderWriterLock ?
   (e) What is ReaderWriterLockSlim ?
   (f) A Comparison of Locking Constructs
   (g) Monitor Class
      i. What are the different methods in Monitor class ?
      ii. What was the problem with lock (try-finally monitor) which is resolved new .Net version ?

iii. Can we give timeout using lock ?

iv. What is Pulse and PulseAll ? What is the difference ?

v. Is pulse internally uses ResetEvent ? If not what it uses ?

vi. If pulse is lost (pulse from one thread occurs before wait is called by another thread) what will happen ? Will the another thread get the pulse or it is lost ? What will happen ?

3. Signaling constructs

   (a) SpinWait

   (b) Conditional Variable in C#.

4. Nonblocking synchronization constructs

   (a) InterLocked

   (b) Volatile

   (c) Memory Barrier

   (d) Lock Free Algorithm

      - Reference: `Lock Free Algorithm`

      Code Samples

3.12.17 Deadlock

- Reference: Deadlock
- Code Reference: Deadlock

1. What is a deadlock ? Explain the scenario(s) where you faced deadlock . How did you resolved it ?

2. How can you detect deadlock ?

3. How can you prevent deadlock ? Ans) Using Leveled lock (see Code Reference).

3.12.18 WPF Threading Model

1. What is thread affinity ? How WPF used it ? What are the advantages ?

(a) What is Dispacher ? What is DispacherObject ? What is Dispacher queue ? MSDN atricle on WPF Threading Model Programming WPF 2nd Ed: Asynchronous and Multithreaded WPF Programming

---

3.13 Memory Management

3.13.1 How .Net identify the object is still referenced ? What are Application roots ? How .Net create reachable graph ?

Ans) Reference to Garbase Collection chapter of book 'Pro .Net Performance'

3.13.2 How to handle memory leak in .Net ?

Capter Memory Leaks from Under the Hood of .NET Memory Management

3.13.3 What is weak reference in .Net ?

Ans: Weak references are a supplementary mechanism for handling references to managed objects. The typical object reference (also known as strong reference) is very deterministic: as long as you have a reference to the object, the object will stay alive. This is the correctness promise of the garbage collector.

However, in some scenarios, we would like to keep an invisible string attached to an object without interfering with the garbage collector's ability to reclaim that object's memory. If the GC reclaimed the memory, our string becomes unattached and we can detect this. If the GC hasn't touched the object yet, we can pull the string and retrieve a strong reference to the object to use it again.

This facility is useful for various scenarios, of which the following are the most common:

- Providing an external service without keeping the object alive. Services such as timers and events can be provided to objects without keeping them referenced, which can solve many typical memory leaks.

- Automatically managing a cache or pool strategy. A cache can keep weak references to the least recently used objects without preventing them from being collected; a pool can be partitioned into a minimum size which contains strong references and an optional size which contains weak references.

- Keeping a large object alive with the hope that it will not be collected. The application can hold a weak reference to a large object that took a long time to create and initialize. The object might be collected, in which case the application will reinitialize it; otherwise, it can be used the next time it is needed.

Chapter on Weak Reference from Pro .NET Performance Chapter on Weak References introduction from Under the Hood of .NET Memory Management Chapter on Weak Reference problems and usage from Under the Hood of .Net Memory Management Code Sample for Weak Reference

3.13.4 If there is a class with two members int and a reference type . Where will int be allocated in stack or heap ?

- Ans) It will be allocated on heap.

- Reference: Memory Management in .Net

3.14 LINQ

- Code Reference LINQ

3.14.1 What is difference between IQueriable and IEnumerable ?

- References: IQueryable vs IEnumerable

IQueryable	IEnumerable
Used for querying databases.	Used for querying in-memory contain
It takes care of the conditions and include in the select query.	Retrieves all the rows in the table do

Refer: IQueryable$_{\text{vsIEnumerable}}$

3.14.2 What is *Expression Tree* in c# ?

3.15 Dynamic Programming Concepts in .Net:

- Code Reference: Samples

3.15.1 What is *dynamic* keyword in c# ?

3.16 Collections/Iterators/Algorithm

3.16.1 What is difference between IEnumerator and IEnumerable ?

3.16.2 What is yield ? What is the purpose ?

Code Reference: Enumerator

3.17 Unmanaged Programming

3.17.1 What is pinvoke ?

3.17.2 What is unsafe keyword ?

Code Reference: pinvoke and unsafe

3.18 WPF/WinForm/GUI

3.18.1 1.2.4

3.18.2 1.2.5

3.18.3 What is Dependency-Property ?

Ans) Dependency Property provides . Change notification . Property value inheritance . Support for multiple providers . Property which is registered with the WPF framework and is available to all the child components. . child can override the property.

3.18.4 What is routed event ?

1. Explain tunnelling and bubbleup event ?

3.18.5 Explain in brief ICommand.

3.18.6 Explain in brief XAML.

3.18.7 Explain in brief Binding.

3.18.8 Explain in brief INotifyChange.

3.18.9 Explain in brief Context.

3.18.10 What is Prism ?

1. What are the functionality Prism provides ?
   (a) What is IOC/DI ? How it is provided in Prism ?
   (b) What is a Loader in Prism ?
   (c) What is a Shell in Prism ?
   (d) What are Events in Prism ? How are they different from .Net Events ?

---

4 Data Structure Questions

- Data Structure Code Samples

4.1 Create queue with two stacks .

- one stack should be used at time in case of (TODO complete it)

4.2 How hash table is implemented internally ?

4.2.1 What are the different ways to resolve collision ?

---

5 Programming Logic and Computer Science fundamentals

5.1 Write a program to give Fibonacci series ?

5.1.1 Write both recursive and non-recursive method for this ?

1. Which is mode optimal in terms of space and time ?

5.2 Write a code to represent int into binary.

5.2.1 Implement it usung bitwise operator ? Explain in detain the flow.

5.3 What is Bitwise operations ?

5.4 What are the different types of Bitwise operations ?

5.5 Why are the advantages of Bitwise operations ?

- Code Reference: Bitwise operation code samples

---

6 Distributed programming and Inter-Process Communication.

- Code Reference: Samples

6.1 What is protobuf ?

6.2 What is microsoft one message bus ?

6.3 Implement a distributed cache.

6.4 What are different inter-process communication methods ?

6.4.1 What is Memory-Map.

6.4.2 What is Shared Memory.

6.4.3 What is difference between Memory-Map and Shared Memory ?

- Code Reference: Cache and Memory Mapped Files

6.5 How socket is implemented in Hermes ?

6.6 What is the difference between Topic and Queue in EMS ?

Topic	Queue
Topic is shared datastructure. Server pulish it into Topic.	Each client has i
Clients fetches the copy from the topic.	Client dequeue it
Message in Topic is removed once all the clients got the copy.	Queue is persista
Topic is transient.	Queue is point-to
If message is sent on topic,who ever subscribed at that time only get a copy each.	In queue case, ea

6.7 Two process sharing a file . Make sure they are synchronized and no one get starved if there is continues flow of data (both for read and write).

6.8 What is Latency ? What is Throughput ? What is the difference between the two ?

- Reference: Latency and Throughput

---

7 SQL

7.1 What are different types of joins ? What is cross/equi/inner/self/non-equi/natual join ?

- Reference: Join

7.2 What are Set commands in Sql ? What is Union/Union All ? What is difference between Union and Union All ?

- Reference: `Set` (Incomplete)

---

8 Algorithms

8.1 Recursion

Recursive
Concise, clear and simple
Expensive: Consumes more time and space: it has to allocate memory on the stack for arguments,
Duplicate calculations: Recursion has more negative impacts on performance if there are duplicat
Stack Overflow: A more serious problem with recursion other than inefficiency is that it causes err

9 Interview Questions Asked

9.1 .Net Lead Poland Interview questions : (December 2013)

9.1.1 3.14.1

9.1.2 `Mutable vs Immutable objects`

1. 1

9.1.3 1.2.1

9.1.4 `IOC`

1. 3

9.1.5 3.7

1. 3.7.1

9.1.6 2.2.3

9.1.7 3.15.1

9.1.8 3.14.2

9.2 Intutive Banglore : Round I (December 2013)

9.2.1 4

9.2.2 5.1

9.2.3 1

9.3 Intutive Banglore : Round II (9th January 2014)

9.3.1 What is External sort? If you have a records to sort which can not fit in the memory ? How you will sort it ?

Ans) External Sort

9.3.2 Design classes for chessboard ?

Ans) Design Chess Program

9.3.3 3.3.1

9.3.4 1a

1. 6a

9.3.5 Is their any issue you handled recently ? Describe .

1. How you handled it ?

9.3.6 What are features in c++11 ?

9.3.7 How Hermes/Agora maintain session ?

9.3.8 How google search works ? How they give options for possible completion ?

9.4 eCommerce .Net Developer Poland Round 1 (December 2013)

9.4.1 What is protofix ?

9.4.2 6.1

9.4.3 6.2

9.4.4 1b

9.4.5 6.3

9.4.6 6.4

9.4.7 6.5

9.5 eCommerce .Net Developer Poland Round 2 (December 2013)

9.5.1 How recovery works ?

9.5.2 Which message format do you use ?

9.6 Tool Development Credit-Suisse Mumbai (November 2013)

9.6.1 6.6

9.7 Morgan Stanley Interview for Equity Derivatives Trader Desktop Team (Position VP) : Round 1 (December 2013)

9.7.1 3.4.1

9.7.2 3.4.2

1. 1

9.7.3 3.13.4

9.7.4 3.5.1

9.7.5 3.6.1

9.7.6 4.1

9.7.7 6.7

9.8 Morgan Stanley Interview for Equity Derivatives Trader Desktop Team (Position VP) : Round 2 (December 2013)

9.8.1 1.2.4

9.8.2 3.3.1

- Given a Sample `* we have Employee class and Manager class . * Manager is derived from Employee. * We have a method which takes a Employee list as parameter.  * Can List of Manager be passed to the above method ? * What if we are dealing with array i.e.  in the above methods we use array and not List ?`

9.8.3 `How .Net identify the object is still referenced ?`

9.8.4 `What is worker thread ?`

9.8.5 `What is background thread ?`

9.8.6 `What is difference between foreground and background ?`

9.8.7 1

9.8.8 2

9.8.9 3.13.2

9.8.10 3.13.3

9.8.11 4.2

1. 4.2.1

9.8.12 1.2.3

9.8.13 1.4

9.8.14 `What is backgroung worker thread ?`

9.9 1e Round I (December 2013)

9.9.1 What are the issues if the key in hashtable is string ?

1. How to resolve that ? How to make better hash function in this case ?

9.10 1e Round II (December 2013)

9.10.1 5.2

1. 5.2.1

9.10.2 1.5

9.11 Amazon (November 2013)

9.11.1 Design a chess program for online gaming.

9.11.2 Tell abount difficult situation or issue you faced ? How you handled it ?

9.11.3 What is your strength and weaknesses ?