

## Technical Interview Questions

### General Software Engineering Questions

- 1. What are the three tenants of object oriented programming?**  
*Inheritance, Encapsulation, Polymorphism*
- 2. What is the difference between boxing and un-boxing?**  
*Boxing is the process of converting a value or primitive type to reference type. Un-Boxing is the process of converting a reference type to value type.*
- 3. What is serialization and de-serialization?**  
*Serialization is the process of taking an objects state in memory and converting it to another format such as XML, binary, stream, etc. for the purpose of transferring across a network or storing. De-serialization is the process of taking a serialized objects state (XML, binary, stream, etc.) and hydrating it back into an object in the applications memory.*
- 4. What are regular expressions used for?**  
*Regular Expressions are used to match specific or fuzzy patterns within a character string.*
- 5. List some of the general differences along with advantages and disadvantages of a web versus desktop application.**  
*Web application benefits include the ability to deploy application logic on a single or group of web servers without the need to install software on a client machine. A web application will be able to target a multitude of devices and operating systems since it relies on http and web standards. Desktop application benefits include a better user experience because there is usually a larger set of controls to work with as well as the ability to interact with specific hardware on the client machine. Desktop applications do not have the interop flexibility that web applications do and the software must be installed on each client machine.*

### .NET Framework

- 1. What is the difference between value and reference types in the .NET framework?**  
*Value types are created on the stack and are passed between classes and methods by value – meaning that another class or method can alter the original state of the value type. Reference types are created on the heap and are passed between classes and methods by reference – meaning that another class or method will only alter a copy of the reference types state.*
- 2. What is the base type that all reference types inherit from?**  
*System.Object*
- 3. What is the base type that all value types inherit from?**  
*System.ValueType*
- 4. What is the difference between a datareader and dataset?**  
*A datareader is a forward only cursor and contains one set or table of data. Because of its forward only cursor it persists a connection to the database until the connection is disposed of. Datareaders are faster than datasets. A dataset can contain several sets or tables of data. A dataset is always disconnected from the database, and you can read data in a table going forwards or backwards.*
- 5. Explain the difference between the BCL (Base Class Libraries) versus the FCL (Framework Class Libraries).**  
*The BCL contains fundamental types including all primitive types such as string, int, long, bool, and basic file access, collections, I/O streams, and garbage collection. The FCL contains types built on the BCL including ADO.NET, ASP.NET, WinForms, WCF, etc.*
- 6. In the context of a LINQ expression, what is the advantage of deferred query execution vs immediate query execution?**  
*In deferred query execution the LINQ expression is executed when the query variable is iterated over, not when the query variable is created. This gives the advantage of modifying the original collection variable and accounting for added or removed items.*
- 7. Explain what Intermediate Language is.**  
*Intermediate language is compiled from human readable source code into a byte language that the Common Language Runtime understands. It is Just In Time compiled before execution into machine language.*

8. **Explain what Just In Time compiling is.**

*Just In Time compilation is the process of taking Intermediate Language assemblies and compiling into machine language just before code execution.*

9. **What are some best practices of exception handling?**

*Have a global exception handler that will catch unhandled exceptions close to the top of your presentation layer components. In lower level code try to handle exceptional code before an error is thrown, but in legitimate cases where you want code to run after a specific exception has been thrown that is acceptable.*

## General Web Development

1. **What are examples of Http verbs?**

*GET, POST, DELETE, PUT*

2. **Since http is a stateless protocol, what are different ways to persist an objects state within ASP.NET?**

*ViewState, Session, Database, State Server, Cookie*

3. **What is the core JavaScript object used to provide ajax functionality?**

*XmlHttpRequest*

4. **What is a css selector and provide examples.**

*CSS selectors are patterns used to select the DOM elements you want to style. Ex: .class, #id*

5. **What are http status codes and provide examples.**

*Http status codes are numeric values returned from the web server to indicate what type of response was issued. 404 - page not found, 200 - success, 500 - internal error*

6. **Explain ways to improve page load speeds.**

*Consolidation of external scripts and style sheets to reduce the number of downloads the browser has to make; Minification of external scripts and style sheets; Using image sprites;*

## ASP.NET WebForms

1. **Explain the ASP.NET page lifecycle.**

*When an ASP.NET page runs, the page goes through a life cycle in which it performs a series of processing steps. These include initialization, instantiating controls, restoring and maintaining state, running event handler code, and rendering. It is important for you to understand the page life cycle so that you can write code at the appropriate life-cycle stage for the effect you intend.*

2. **How can you achieve user interface re-usability?**

*The use of a user control.*

## ASP.NET MVC

1. **Describe the purpose of each component in the ASP.NET Model View Controller framework.**

*The model contains application state and business logic for the application. The controller contains application logic and creates the necessary model and view for the request. The view contains markup/logic specific to the user interface for displaying data from the model.*

2. **What are the benefits and drawbacks of MVC over WebForms and vice versa?**

*MVC Benefits:*

- Clear separation of concerns in regards to where logic resides in the application.
- Controllers and models very easy to unit test.
- More control over rendered markup/script in the browser.
- Absence of view state which makes pages more responsive.

*MVC Drawbacks:*

- Longer development time.
- Lack of event driven model that WebForms has.

*WebForms Benefits:*

- Rapid application development.
- Event driven/postback model
- Ideal for prototyping.

#### *WebForms Drawbacks:*

- Viewstate can get very large and affect page responsiveness
- Do not have 100% control over rendered markup from web controls.
- Not easily unit testable.

3. **What is the purpose of an ASP.NET MVC HTML Helper and how would you implement one?**

*An MVC Helper contains re-usable server side code that aids in the rendering of an MVC view. HTML helpers are implemented via extension methods. The most common object that a helper extends is the MvcHtmlString, but you can extend any object on the server side to create HTML helpers.*

4. **What is an ASP.NET MVC Area?**

*Areas are used to organize an MVC application into vertical tiers. Each tier represents a common set of functionality -- such as admin, reporting, and shopping.*

5. **What scenarios would it be best to use an HTML helper or partial view and vice versa?**

*An HTML helper is best used when rendering HTML based upon server side logic. Since HTML helpers are extension methods -- it is easy to pass information from the model to the helper to generate HTML. A partial view is best used for re-using html and UI specific markup.*

6. **Explain what an ActionResult is and list some implementations and use cases.**

*An ActionResult is an abstract object that represents the return type of a controller action method. Concrete implementations include: View - which returns a defined HTML view, JSON - which returns a .NET object that has been serialized into Javascript Object Notation, Redirect - which returns a redirect to another URL, File - which returns a file stream, etc.*

7. **Explain what an MVC Filter is and it's use cases.**

*An MVC filter is a set of cross-cutting functionalities that can be used throughout an application. It is implemented using attribute based programming and examples would be security, error handling, etc.*

#### **WCF**

1. **Define Service Oriented Architecture.**

*A system that utilizes SOA is one that packages alike functionality into a set of interoperable services that can be used within multiple, separate systems from several business domains.*

2. **What are the most interoperable bindings in which all other platforms would be able to consume service methods?**

*WebBinding, BasicHttpBinding, WsHttpBinding*

3. **What are the three main ways to host a WCF service?**

*Self-hosted, IIS, WAS(Windows Activation Service)*

4. **What are some of the tools that can be used to code generate service proxies?**

*WSDL.exe, SVCUTIL.exe*

5. **What are the ABC's of a WCF endpoint and give a description of each?**

*Address specifies the location of the service in which a service consumer will send requests. Binding describes how the service and client will communicate in terms of transport mechanism, encoding and protocols. Contract specifies the interface between client and service.*

6. **What would be the best way to expose a service on the same intranet to a Java client and .NET client simultaneously?**

*Expose 2 different endpoints; A TCP binding that .NET clients can connect to and an interoperable endpoint such as HttpBinding/WsHttpBinding for non-.NET clients.*

7. **List the advantages and disadvantages of consuming WCF services via channel factory, code generated service proxy, or manually built proxy.**

*Using the channel factory will rely on classes defined within a shared contract assembly that must be updated if a service/data contract is modified on the service. A channel factory can only be used by other .NET clients and is not interoperable. Service references or code generated service proxies must be re-generated each time a service/data contract is modified and is vulnerable to duplicate proxy class definitions between service endpoints. Manually built proxies offer the greatest flexibility, but must be manually modified each time a contract is updated.*

## Algorithms

1. **Define an algorithm that will reverse a character string.**  
*Split the string by characters and place into an array, then read the array last to first.*
2. **Define an algorithm that will iterate through a hierarchal file system and delete only text files.**
  - a. *Create a recursive method that accepts a root folder on the file system as a parameter.*
  - b. *Within the method write a loop that will iterate through each folder within the root folder parameter and pass children folders to the original method.*
  - c. *Within the loop check each file extension and if it's a text file delete.*
3. **Define an algorithm that will return a value indicating if a string is a palindrome.**  
*Reverse the string and then return the value comparing for equality against the original.*

## Design Patterns

1. **List the 3 design pattern categories originally identified by the Gang of Four.**  
*Creational, Behavioral, Structural*
2. **List examples of design patterns that are baked directly into the .NET Framework.**  
*ForEach loop => Iterator Pattern*  
*Events/Delegates => Observer Pattern*  
*Extension Methods => Decorator Pattern/Fluent Interface*  
*TreeNode Controls => Composite Pattern*
3. **What is the difference between the Abstract Factory and Factory Method patterns?**  
*Abstract Factory provides an abstract interface for creating a family of objects. Factory Method provides an abstract interface for creating one object.*
4. **Explain the purpose of a Facade design pattern.**  
*A facade is used to hide and unify the complexity of a group of components behind one API.*
5. **Explain what a fluent interface is.**  
*A fluent interface is one which utilizes method chaining to relay the instruction context of a subsequent call.*