



.NET Developer Technical Interview Question & Answers!

Contents

General Programming (C# focus)	1
HTML/CSS/JavaScript (front-end web development)	4
SQL and Relational Database Design	7
ASP.NET.....	8
ASP.NET - MVC	10
ASP.NET - Web Forms	12

General Programming (C# focus)

1. What is object-oriented programming (OOP)?

- Uses classes/data types
 - to model/map to real-world things/objects/concerns
 - addressed by our program.
- Accounts for characteristics (properties) and behavior (methods)
- Classically supports inheritance, polymorphism and encapsulation.

2. What is a class?

- Basic building block of code in C#
- Often used to define datatypes but can be a
 - repository for methods or properties (warehouse)
 - or represent a program like with Console program's Main()
- Blueprint/Template of properties or methods for actual objects

3. What is an object?

- One instance/variable of a type/class/struct
- Has its own state (values)

4. How is method overriding different from method overloading?

- You can have a child class override parent class methods
- That are marked as virtual or abstract
 - Changing how it works for the child
 - Otherwise you get what you inherit
 - (In case of abstract, you HAVE to provide the override definition in the child because abstract parent just has a name, no functionality defined)
- Method Overloading is defining the same method to work in different ways based on different signatures/unique parameter sets.

5. Why is the virtual keyword used in code?

- To mark a member/method as "overridable" by child classes of this class.

6. Explain the concept of constructor.

- Specialized method to create objects of the complex type where the ctor (constructor) is defined.
- Same name as type where it's defined
- **new** keyword to call it
- Needed because unlike simple types, like string/int/bool – no way to know how to give values to various custom properties of the type at variable creation.
- Also lets us decide what gets set and how, with different ctor overloads.

7. What is the syntax to inherit from a class in C#?

```
public class Childclass : Parentclass {}
```

8. What are methods?

- Reusable blocks of code
- Perform actions
- Verbs of the language.
- Like "functions" or "subroutines" in other languages.

9. What is a namespace? Can you name one you've used from the .NET framework?

- Logical container for related classes.
- Sort of like folders for your reusable code (classes)
- Examples:
 - System, where Console, Math and Random are defined for working with those topics.
 - System.Net.Mail for email
 - System.ComponentModel.DataAnnotations for MVC model validation and UI hints.
 - System.Web.Configuration for namespace containing classes related to manipulating the Web.Config file.

10. What is the function of the Try-Catch-Finally block?

- Runtime Exception handling
- TRY segment runs the "dangerous code", but with the option to catch and handle a runtime exception if that's what would have occurred at runtime otherwise.
- CATCH blocks stop the program if they TRY's block would have caused an exception, and instead "do something else". You can have multiple CATCH blocks to respond to different kinds of exceptions.
- FINALLY blocks run after either TRY or CATCH blocks have run, for a "and now, let's do this" option.

11. What are abstract classes? What are the distinct characteristics of an abstract class?

- Classes that force overriding
- They set common shared method names, but do not provide a base functionality – it must be determined by child classes

12. What is inheritance in OOP?

- The ability to create child classes that automatically include properties and methods defined in a parent class.
 - Increases code re-use and maintainability.
 - Allows the parent class to define some methods as virtual or abstract, thereby allowing child classes to override how those methods are implemented differently by the child.
 - Inheritance chains allow for a series of parent-child relationships, but no child can have more than one direct parent.
 - Interfaces provide a "must-implement" or "is a kind of" relationship where a class can implement many interfaces (which allows greater flexibility than just true parent-child inheritance relationships).

13. What is the difference between containment and inheritance in OOP?

- Inheritance is sometimes called an "IS A" relationship.
 - A Customer (child) "is a" Person (parent).
 - An HourlyEmployee (child) "is an" Employee (parent).
- Containment is sometimes called a "HAS A" relationship.
 - A Sale "HAS A" Customer as a property.
 - An Order "HAS A" collection of Product objects as a property.
- Inheritance allows us to create a class that automatically includes the properties and methods defined in the parent class.
- Containment allows us to define the property of a class that is itself a complex class.

14. What is polymorphism in OOP?

- Closely tied to inheritance, polymorphism is the ability of a child-class object to be boxed as a parent-class object, but when one of its “parent-defined” methods is called, the child-class overridden behavior is used at runtime.
- Example: Parent class is Employee, which defines a virtual method CalculatePay(). HourlyEmployee is a child class of Employee and it overrides the CalculatePay() method to work differently for the child class. This alone is not polymorphism. Polymorphism is when an HourlyEmployee object (child) is boxed or stored as a parent-class object (Employee), and then has its CalculatePay() method called. It can be called even though the child object is stored as a parent object because it was defined in the parent. BUT (because of polymorphism and runtime implementation) that child class object stored as a parent class object will have its **overridden** method called at runtime.

15. What is encapsulation in OOP?

- Encapsulation is the act of “hiding” code functionality in another container – the caller of the functionality can use the code but has limited control over how the code does its job.
- Example: calling a property of a class allows us to GET or SET the field’s value, but how that getting or setting is done is hidden and handled from within the property method logic.

16. What is Language Integrated Query (LINQ)?

- SQL-like syntax added to .NET languages for manipulation, analysis, sorting/filtering of different types of data, including
 - .NET collections – called LINQ to Objects
 - Relational databases – LINQ to SQL –or– LINQ to EF (Entity Framework)
 - XML – LINQ to XML
- Benefits
 - Relatively easy to learn
 - Allows for Intellisense, build errors, color coding
 - Usable against multiple types of data sources
 - Performance boost – only processed when LINQ variable is used.

HTML/CSS/JavaScript (front-end web development)

17. What languages does a browser understand and what are the purposes of each?

- HTML – content
- CSS – display
- JAVASCRIPT – extended functionality (validation, AJAX calls and JSON interpretation, UI enrichment)

18. What are three ways that we can include CSS in a website? Which has the highest precedence? Which is the recommended technique?

- Inline, internal, external
- Inline has highest precedence
- External style sheets have the highest re-usability and are best practice.

19. What are the three main types of CSS selectors? Whiteboard out referencing one of each in a style sheet.

- ID (#), class (.), element (tag name)
- ```
#idName{}
.className{}
tagName {}
```

**20. What is the basic structure of an HTML page?**

- DTD, html contains all else, inside it are header and body
- Header is for metadata and links to other files (CSS/JS) the page needs
- Body is for content of the page

**21. What CSS property changes the space between the content and border of an element?**

- padding

**22. What CSS property changes the space outside of the border of an element?**

- margin

**23. If you want to float a block element to allow other content to flow beside it, what css property other than float should you change?**

- width

**24. You come across a parent container in your HTML. It contains a floated child container, some other content, and a child container at the bottom with a “clear fix”. Why?**

- The content in floated elements does not count towards the height of the parent container like normal HTML rendering.
- So if the floated element is taller than its container, it will extend outside of the parent instead of being properly contained.
- Adding a tag with `clear:both` at the bottom of the parent will force the parent to be at least as tall as the floated object above.

**25. What are the three main image types that are used on websites?**

- JPG / JPEG
- GIF
- PNG

**26. Which two measurement types should be used for a Responsive Web Design?**

- % for widths and em for font-sizes, generally

**27. In CSS, what is used to specify rules that only apply when a screen changes to a certain size?**

- Media query

**28. Whiteboard out a CSS specification that the container with an id of wrapper should be 90% width and centered when the viewport is 500px or thinner.**

```
@media screen and (max-width: 500px) {
 #wrapper {
 width: 90%;
 margin: auto;
 }
}
```

**29. In JavaScript, how would you select an html element with the id of FavoriteFood?**

```
var ff = document.getElementById('FavoriteFood');
```

**30. What is the doc ready function in jQuery?**

- It is the standard container (event handler) for embedding all custom jQuery code. It is more efficient than using the JavaScript `pageLoad()` because it starts processing the embedded code after only the most important page components are done loading (instead of waiting for assets like images to load).
- It can cause conflicts with Web Forms partial page updates.

**31. Whiteboard the jQuery shortcut for `$( document ).ready( function() {} );`?**

```
$(function() {
 //function code here
});
```

**32. In jQuery, how would you select an html element with the id of FavoriteFood and apply a class of "food"?**

```
var ff = $('#FavoriteFood');
ff.addClass('food');
OR
$('#FavoriteFood').addClass('food');
```

## SQL and Relational Database Design

### 33. What is a primary key?

- One or more fields in a table that uniquely identify the record in the table

### 34. What is a foreign key?

- A field in one table that associates the record with a record in another table.
- The foreign key in one table maps back to a primary key field in another table.

### 35. What is an aggregate function? When would you use one?

- An aggregate function takes a collection of records and applies a calculation against all records' values, returning exactly one answer.
- It can be broken down into smaller subsets of results using GROUP BY clause.
- Those subsets can be further filtered with a HAVING clause.

### 36. When do you need an outer join?

- (Default) Inner Joins will only include records related across both joined tables.
- If we need to include records even when they are not related to a joined table, we use an outer join.
- Example: If I use an inner join to select all records between products and categories, but some products don't have categories, they will be excluded. To include all products even if they don't have a category, we could do LEFT, RIGHT or outer join.
  - If the foreign key to the categories table is a required field, this isn't needed – all products will have a category.
  - If we want to get all categories and products, including categories never used by a product, we're back to using some form of outer join.

### 37. How would you design a database for a car dealership where we want to track cars being sold to 1 or more customer at the same time (co-owners)? Whiteboard out a design for one or more tables tracking the cars, sales, and customers.

- This will require a many-to-many relationship since a sale of a car can have multiple customers associated.
  - Cars table
    - PK CarID
    - Make
    - Model
    - ...
  - Sales table
    - PK SaleID
    - DateSold
    - Price
    - ...

- Customers table
  - PK CustomerID
  - FirstName
  - LastName
  - ...
- Customers\_Sales bridge table
  - PK SaleID
  - PK CustomerID
- The bridge table will have a record for each sale being associated with one particular customer, allowing you to have as many customers as needed per sale.

**38. Imagine a database with a Customers table. Write a query that returns the CustomerID, Name, City and State for only customers with a Name starting with 'M'. Sort on State then City.**

```
SELECT CustomerID, Name, City, State
FROM Customers
WHERE Name LIKE 'M%'
ORDER BY State, City
```

**39. Imagine a database with a Products table. Write a query that returns the highest product price broken down by vendor. Only include prices higher than \$1,500. Sort by highest prices first.**

```
SELECT VendorID, MAX(price) AS [Highest Price]
FROM Products
GROUP BY VendorID
HAVING MAX(price) > 1500
ORDER BY MAX(price) DESC --OR [Highest Price] column alias
```

## ASP.NET

**40. What is ASP.NET?**

- Microsoft's modern/latest answer to "dynamic web pages/applications".
- Can be
  - Web Forms (designer, smart UI with ViewState, asp.net page lifecycle, server-side events)
  - MVC (More modular & testable, better HTML/CSS, works more easily with jQuery)
  - Web Services / Web API (programs that expose methods over the Internet that return data for other programs to use. Have no UI – that's the calling program's job.)



**41. What is IIS? Why is it used?**

- "Internet Information Services"
- Microsoft's web server to host/serve up web pages
- Has built-in .Net compatibility, so it can process ASP.NET web pages.
- Other web servers, like Apache, can't do that by default.

**42. What is a Query String? What are its advantages and limitations?**

- Widely supported and long-standing technique for passing info in the Page Request that can be used by the requested page for dynamic server-side processing.
- Examples: to change the page's content, send an email or update a database about the QS content.
- Used by lots of web technologies (Java JSP, PHP, ASP, ASPX, etc)
- Limited to passing only STRING info (not types like DateTime, Exception, etc)
- Also, browsers may limit the #characters allowed
- And info is not private – shows up in address bar
- Can be a good thing, being able to copy and paste the link with querystring variables for re-use.

**43. What is the difference between authentication and authorization?**

- Authentication = ID
- Authorization = permissions

**44. How can you send an email message from an ASP.NET web page?**

- Use System.Net.Mail namespace for MailMessage and SmtpClient classes
- Create MailMessage object with from/to/subject/body fields
- Create SmtpClient object with address of the SMTP server where we have permissions to send.
- Call client's Send() on the MailMessage object

**45. What is a data access layer (DAL)?**

- A set of code to promote re-use of common data functionality in your program, including:
  - Accessing a database
  - Selecting one or more records and translating them into OOP objects.
  - CRUD actions to modify data in the database from your program

**46. What is a business logic layer (BLL)?**

- A set of classes (domain models) that can be mapped to records in a database.

**47. What is an ORM? What is an example you've worked with?**

- An Object Relational Mapper is a framework that provides a DAL (data access layer) and a BLL (business logic layer) for working with data in your program in a strongly typed way.
- Entity Framework (EF) is the ORM we used in the Centriq Track.

**48. What is role-based security?**

- Being able to put users into roles and then lock down the folder or controller accessibility (and/or page content visibility) based on the role of a user.

## ASP.NET - MVC

**49. What are the 3 main components of an ASP.NET MVC application? What are their roles?**

- Model: classes that model the data used in our application.
- View: HTML, CSS, JavaScript and Razor code that represents our webpages.
- Controller: a class that handles the request and response process, and passing data for views and models.

**50. What are the 3 segments of the default route that is present in an MVC application?**

- Controller/Action/ID

**51. What is an action?**

- A specialized method in a controller that handles requests and generally returns a view.

**52. Do actions always return views of the same name? Explain.**

- No. Actions can return views of the same name or any other view as needed.
- For example, a standard GET action will often return a view of the same name, but a POST version of that action might return a confirmation page after collecting user input from a form.

**53. What does the phrase "Separation of Concern" mean and how does it relate to MVC?**

- Separation of concerns means breaking out your code to handle individual responsibilities.
- The goal is more modular, re-usable and loosely-coupled code.
- In MVC, you will see:
  - Models define types of common data, including business rules and validation.
  - Views define the presentation layer – pages for display and collection of data.
  - Controllers handle the request and response cycle.

**54. What types of views can you use to maintain shared content used by multiple views?**

- `_Layout`: represents a shell of common web page content that other views can use to “wrap around” their own custom content.
- `PartialView`: represents a common set of web page content that can be “plugged in” on as many views as needed.

**55. What code segment is used on the shared layout page to denote where the specific view content is to be included?**

- `@RenderBody()`

**56. What is Razor?**

- A syntax for .NET languages to be used in the view (nested in HTML/CSS/JavaScript code)
- NOT a language in its own right.
- Allows plugging in dynamic, server-side content

**57. What is needed to specify validation rules within a model?**

- Add namespace  
`using System.ComponentModel.DataAnnotations;`
- Add necessary data annotations attributes above appropriate properties, such as:  
`[Required]` or `[StringLength(100)]`

**58. What symbol would you use to denote the use of Razor?**

- `@`

**59. What is needed create a GET and POST versions of the same action?**

- The action method is defined twice (overloaded).
- So each version must have a different signature (parameter set).
- The POST version must have the `[HttpPost]` attribute written above it. (The `[HttpGet]` attribute is implied by default for the GET version.)

**60. Talk to me about the ViewBag in MVC.**

- We can use it as one way to pass data from the controller to the views (or between views, like passing data from a view to the `_Layout`).
- `ViewBag` variables are dynamic data, meaning the data type is implied upon creation based on the data assigned. You do not need to unbox or convert the value when extracting it (unlike some other tools like `Session` or `QueryString`).

- ViewBag variables are not supported by Intellisense – you must manually call on them and if done incorrectly, the error will be seen at runtime.
- Usually used for small amounts of info. Using strongly typed data to pass more complex values is often preferred.

#### **61. What does it mean to use strong types in MVC?**

- Instead of passing information between the views and controllers as small, simple values or ViewBag variables, we can pass one specific type between a controller action and the view being returned.
- Strong types are supported by Intellisense, color-coding and compiler error-checking in Visual Studio, as opposed to ViewBag variables.
- Example 1: instead of passing 5 different ViewBag variables to a view, we could pass one object of a strong type (model) that has those 5 values as properties.
- Example 2: instead of collecting 12 different individual values from the user in a form, we could create a model that defines those 12 values as properties. The view would be strongly typed to that model so that when the form is posted, we can collect the one strongly-typed object and can easily use it with Intellisense.
- Using strong types also allows us to use scaffolding of views or even controllers, actions and views.

#### **62. What is the file extension for C# Razor views?**

- .cshtml

#### **63. What is meant by the phrase “n-tier architecture”?**

- Separating a solution’s code over more than one project.
- The goal is a separation of concerns, which should also lead to greater code re-use and interoperability.
- Examples: instead of having one solution that handles the UI, data access and data models for various databases, it could be broken out into separate projects that work together. So, one or more projects handling data could be re-used for multiple different “program” projects (user interfaces).

## **ASP.NET - Web Forms**

#### **64. How many forms should every Web Forms page contain?**

- Exactly one, with runat="server", to track ViewState between postbacks.

#### **65. What is a postback?**

- It’s a round trip (request and response cycle) that is re-processing the current page.
- ViewState is used to track the current state of all the controls on the page.

**66. In which page event are the controls fully loaded?**

- All controls are loaded by the Page\_Load event of the page events lifecycle. It has been noted that controls are accessed in the Page\_Init event however the view state is not loaded at this point.
- Direct Answer: All controls are fully loaded at the Page\_Load event."

**67. How can we identify that the Page is Post Back?**

`if(Page.IsPostBack)`

OR, page implied:

`if(IsPostBack)`

In the code-behind

**68. What is ViewState?**

- Collection of info for the state of the controls and page
- Maintained between postbacks of the page
- Data encrypted on HTML page
- Passed between each Request/Response cycle

**69. What is the lifespan for items stored in ViewState?**

- Stored upon initial page load, and then maintained while page continues to be posted back (reprocessed).
- ViewState is lost as soon as you leave the page by requesting something else
- Returning to the page causes ViewState to reset again – going through "initial page load" again

**70. Explain login controls.**

- Used for web application membership
- To restrict access of folders and visibility of page content.
- Login – to allow users to login with their passwords.
  - Required for web forms authentication.
  - Should be on the Login.aspx or another specified default login page
- LoginStatus – provides a Login link to the default login page when anonymous and Logout option when logged in
- LoginName – displays user's username when logged in
- ChangePassword – lets user update their password
- PasswordRecovery – lets user reset their password
- CreateUserWizard – lets anonymous users create accounts

**71. What setting must be added in the configuration file to deny a particular user from accessing the secured resources?**

```
<authorization>
 <deny users="username" />
</authorization>
```

**72. What setting must be added in the configuration file to deny unauthenticated users from accessing the secured resources?**

```
<authorization>
 <deny users="?" />
</authorization>
```

**73. What setting must be added in the configuration file to deny all but users in the role of Admin from accessing the secured resources?**

```
<authorization>
 <allow roles="Admin" />
 <deny users="*" />
</authorization>
```

**74. What is State Management? How many ways are there to maintain a state in .NET?**

- Techniques for storing information in a web application without using a database.
- 3-7 depending on definition
  - Technically defined as SM: ViewState, Session state, and Application state
  - Other techniques for persisting data: Cookies, Querystring, Crosspage Postback, Server.Transfer()
  - Additional possibilities include: Control state (like for gridviews), hidden fields, Profile properties

**75. Explain the validation controls. How many validation controls are in ASP.NET 4.0?**

- Mostly client-side (Javascript) except custom validator
- There are 5
  1. Required Field Validator
  2. Regular Expression Validator
  3. Compare Validator
  4. Range Validator
  5. Custom Validator

**76. What is the function of the CustomValidator control?**

- Mostly as our "all-else" validator.
- Can be used to create validation in C# if custom code is required and C# is stronger suit or more reliable than JavaScript.
- Allows for validation related to server-side criteria – like comparing a new choice of username against a database containing existing values to ensure uniqueness.
- Only built-in validator to require a server-side event: OnServerValidate()

**77. Which data type does the RangeValidator control support?**

- It defaults to String, but also allows for Integer/Double/Date/Currency

**78. What is the difference between a HyperLink control and a LinkButton control?**

- Hyperlink
  - just takes you to another page/resource.
  - But as a server control it can have various properties like NavigateUrl and Text dynamically manipulated in the code-behind.
  - It is just output as an <a> tag.
- LinkButton
  - is functionally a button, causing a postback when clicked
  - and capable of server-side event handlers like Click events for processing some code.
  - It looks like a link and does have an <a> tag output with JavaScript to make it function like a button.