SQL

1. Function vs stored procedure
2. Query commands
3. Index
4. How would you write an SQL statement to perform an outer join between two tables?
5. Why use an abstract class and not just a regular class?
6. What is your experience in coding?
7. What are the different types of SQL joins?
8. Write an SQL statement to match specific columns of one table to specific columns in another.
9. What is a primary key?
10. What is a foreign key?
11. What is a unique key?
12. What is the identity keyword in SQL?
13. What are the basic SQL constructs?
14. Tell me in an employee database, what will be the basic tables you would have?
15. Apart from employee names, what are key information needed to store in an employee database?
16. Let says you have two tables: Table 1 has employee roles (i.e., Junior/Senior developer). Table 2 lists salaries by roles. Let say you have to find the basic salary for a junior developer. What kind of relationship would you like to build to get that data?
17. Do you know what is a primary index for a table?
18. Tell me about primary key.
19. Can you have multiple primary keys for a single table?
20. Can you have multiple unique keys (for a single table)?
21. What is Truncate and Delete commands in SQL?
22. Tell me about how you can find all people who are Aged 30 under a table named XYZ?
23. Describe composite, foreign, and primary key relations.
24. What are some datatypes in SQL?
25. Give some examples of query commands.
26. What are the SQL commands used to access a table?
27. Have you created a table?
28. What do you think the difference between truncate and delete statement?
29. What are the different joins you used?
30. What do you think the difference between left join and left outer join?
31. What is normalization in database?
32. What are relational databases?
33. What is the difference between char and varchar?
34. What is the cross join
35. What is the difference between drop and delete?
36. What is Trigger in SQL?
37. What is index in SQL?
38. What are truncate and delete?
39. What is a stored procedure?
40. What is an SQL command object?
41. How do you create a database?
42. Give us some examples of query commands?
43. If there are two tables to use, how do you reference them?

SQL

------------------------------------------------------------------------------------------------------------------------------------------------

ADO.NET/ Entity Framework

1. Connection differences between ADO.NET and Entity Framework
2. How you modeled a database (or class) using Entity Framework
3. Error and exception handling in Entity Framework
4. What is ADO
5. How to connect with Database. (ADO)
6. How would you write SQL queries with entity framework?
7. What is Entity Framework?
8. How do you make queries to EF
9. Entity Framework
10. Code first vs. Database first
11. Connecting database to API, how to make changes in the database
12. When using ADO.NET, how do you map the response to the model?
13. What is the difference between ADO and EF

ADO.NET/ Entity Framework

------------------------------------------------------------------------------------------------------------------------------------------------

OOP

1. What are the principles of OOP? Describe each of them.
2. What is the benefit of inheritance
3. What is a class?
4. What is an object?
5. Immutable objects – how to change immutable objects
6. Describe how you would use Polymorphism in one of your projects?
7. What is Overloading? What is Overriding? When would you use these and why?
8. Encapsulation
9. What are abstract methods?
10. What is Interface?
11. What is inheritance and why use it?
12. What is the difference between overload and override?
13. Give me an example of how you have used abstraction in one of your projects?
14. When is abstraction necessary?
15. What is overloading?
16. What is overriding?
17. Encapsulation
18. Polymorphism
19. Run-time Polymorphism and Compile-time Polymorphism
20. What are the OOP pillars?
21. What is method overloading?
22. What’s the difference between Abstraction and Polymorphism?
23. What are the two ways of inheriting something

OOP

------------------------------------------------------------------------------------------------------------------------------------------------

.NET

1. What is the Jit compiler?
2. What is the CLR?
3. Describe to me how you would use references in a project?
4. What is a DLL?
5. What is Common Type System (CTS) in terms of .NET framework?
6. What is an Assembly in .NET?

.NET

------------------------------------------------------------------------------------------------------------------------------------------------

C#

1. What is a constructor?
2. What are Value vs Reference Types in C#?
3. What is Exception Handling and how would you use it? Try Catch? Throw?
4. What is singleton
5. How would your empty variable memory when you no longer need it?
6. What are using statements?
7. What is the difference between authentication and authorization?
8. What forms of user input data validations have you used?
9. What is the difference between error logging and debugging?
10. What access modifiers do you know?
11. What are SOLID principles?
12. How do you handle exception?
13. What are abstract classes and interfaces, and how do they differ?
14. What are constructors?
15. What is Try and Catch?
16. How do you handle an exception?
17. What are abstract methods?
18. What is method overloading and overriding?
19. What is value type vs reference type?
20. What is boxing and unboxing
21. Compare Abstract vs Interface
22. What is LINQ?
23. What is a lambda expression?
24. Describe Int.
25. Describe strings.
26. What's the difference between int and string?
27. Multiple inheritance in C#
28. How to achieve multiple inheritance in C#
29. Do you have any experience with LINQ to ADO.Net
30. How would you use a connection string without hard coding it into your program?
31. What are types of constructors?
32. How do you handle exceptions
33. Can you have multiple catch blocks
34. Should you have exception handling(try-catches) on all levels of your code, what might be the issues?
35. How many ways can you catch Exceptions?
36. What is one specific way of catching exceptions in C#?
37. Have you used Exception in C#? Process of Try Catch?
38. Difference between a value and reference type?
39. How to connect SQL to C#?
40. What is the difference between a string and a String Builder?
41. What is the difference between a struct and a class?
42. What forms of user input data validations have you used?
43. "let's talk about C#, what data types do you know in C#
44. What is the class used to execute from the database?
45. What is dependency injection?
46. What is ADO.NET?
47. What are value and reference types?
48. What are the types of constructors?
49. How would you make sure that a class could not be inherited?
50. Global variable vs Static.
51. What are dependencies?
52. What is error logging?
53. What are micro services/systems? What is data abstraction?
54. What is dependency injection
55. How to connect SQL to C#
56. If the memory is too big, how do you clear that exception with using try/catch block?
57. Is method overloading being a runtime method or compile time method?
58. There are how many types of constructors in C#
59. What is IDispose
60. Have you done any Serialization, What kind?
61. What is finalize and dispose
62. Should you have an Idisposable in all parts of your code.
63. How would you create an employee class?
64. What is IEnumerable
65. What are the different types of constructors

C#

------------------------------------------------------------------------------------------------------------------------------------------------

ASP.NET

1. What is ASP.NET?
2. What is ASP.NET MVC?
3. What is REST?
4. What is the purpose of unit testing?
5. What are some of the segments of software testing before it goes to production?
6. Http methods (what they do)
7. Http request code (200, 500….)
8. What is CORS?
9. How do you connect API to the database?
10. How do you use authentication in ASP.Net?
11. How do you connect a REST API to an SQL database?
12. How do you set up a REST API?
13. How do you connect to the database with an API?
14. What's ASP.NET?
15. What's ASP.NET MVC?
16. How to use Routing in C#?
17. Difference between a post and put?
18. What are the different types of response codes in API?
19. How to create an API?
20. There are how many types of Authentications?
21. What is the difference between Authentication and Authorization?
22. How do you close a connection from a deployed API, if it were attack maliciously?
23. What is a reject?
24. What is Razor and a .cshtml file?
25. Can you have more than one model in a view?
26. What is cashing?
27. What is state management in dot net?

ASP.NET

------------------------------------------------------------------------------------------------------------------------------------------------

GIT

1. Proper coding etiquette, when to check in with your code?
2. What is a branch?
3. code line or stream in regard to source code version control:
4. Is it possible to merge two or more branches in your local environment?

GIT

------------------------------------------------------------------------------------------------------------------------------------------------

Angular

1. What are directives in angular?
2. What is routing?
3. What is the HTTP response codes?
4. What is the difference between let and var?
5. How do you style a selector in CSS file?
6. How to connect Angular to the API?
7. Angular routing
8. What is the difference between let and var?
9. What is session? How long is a session stored?
10. Write out a HTTP request from angular that would connect to your API
11. What means to seal a project
12. What ng you use to bind information
13. What are some angular directives?
14. What is ng Bind?
15. How would you bootstrap an Angular project?
16. How do you add components in Angular?
17. How do you add styling?
18. How do you build angular for hosting?
19. Have you done Angular unit testing?
20. What is lazy loading in Angular?
21. How do you deploy a project in angular?
22. What is the difference between structural and attribute directives?
23. Through Angular, how do you make calls to the backend API?
24. Tell me how you would add a new component to an Angular project?
25. What is bundling? (In angular)
26. What are Decorators and Observables?
27. What are directives?
28. Describe one of the named directives.
29. How do you bootstrap angular project?
30. How would you pass components from to another?
31. How do you deploy a project in Angular?
32. What is routing?
33. What is bundling?
34. What is minification?
35. What is NPM?
36. What are decorators and observables?
37. How to connect Angular to the API?
38. What is the lifecycle hook
39. What is the root module of an angular app
40. Where is the entry point of an angular app
41. How do you bundle angular
42. What is early binding and late binding?

Angular

------------------------------------------------------------------------------------------------------------------------------------------------

HTML/CSS/JS/TS

1. Tell me about how you can change the color of letters under CSS?
2. What is AJAX and have you used it in a project before?
3. What is typescript?
4. How do you compile typescript into JavaScript?
5. JavaScript vs TypeScript
6. What is JSON?
7. What is the difference between let and var?
8. What are promises in JavaScript?
9. How do you handle exception in JavaScript(?)
10. What message will be shown if you print “var x;” in JavaScript? Console.Log(x);
11. What message will be shown if you print “y” that you have not declare that yet?
12. What is Closure in JavaScript?

HTML/CSS/JS/TS

------------------------------------------------------------------------------------------------------------------------------------------------

Unit Testing

1. What is the purpose of Unit Testing?
2. What is Mocking?
3. How would you write a unit test?
4. What is test driven development?
5. What is MOQ?
6. How did you implement unit testing?
7. What forms of unit testing framework did you use?

Unit Testing

------------------------------------------------------------------------------------------------------------------------------------------------

Projects

1. Tell me about configuration in your projects?
2. Explain the steps you took into creating one of your projects?
3. What is your role in the latest project you were doing?
4. Being <this role> what were you some challenges you faced and how did you overcome it?
5. Tell us about your projects in Revature.
6. What’s experience do you have in your projects? What were they?
7. What is SCRUM?
8. What technologies interest you? What is your favorite technology?

Projects

------------------------------------------------------------------------------------------------------------------------------------------------

DevOps/Cloud/Docker

1. API and web app deployment utilizing Azure
2. DevOps experience in GitHub (conflicts and merge errors in particular)
3. Setting up SonarCloud
4. How do you deploy an angular project?
5. Have you used GitHub with Visual Studio (Code) integrated tools?
6. Have you used CI/CD pipeline?

DevOps/Cloud/Docker

------------------------------------------------------------------------------------------------------------------------------------------------