

Senior Software Developer Interview Questions And Answers Guide.



Global Guideline.

<https://www.globalguideline.com/>



Senior Software Developer Job Interview Preparation Guide.

Question # 1

Tell us why are you interested in working at "XYZ Company"?

Answer:-

Show you've done your homework and that you have researched the company. Express your sincere desire to work at said company - but keep it balanced and don't be sycophantic or overly enthusiastic and saccharine.

[Read More Answers.](#)

Question # 2

Tell us what are you looking for in an opportunity that would lure you away from your current employer?

Answer:-

Be frank, but don't just make it all about the money and the benefits. Obviously, these are important elements, and any recruiter knows that those two are key components in your decision-making process. But talk about workplace culture, creative fulfillment, the satisfaction of solving real-world problems etc.

[Read More Answers.](#)

Question # 3

Tell us what is your process for finding a bug in an application? How much time do you typically spend on debugging?

Answer:-

The first question tests the way the candidate thinks when working with difficult bugs. Every candidate has their own process, but they must use a debugging tool, understand how to sift through each line of code using that tool, and then understand what must be done to fix the bug without affecting other code within a project. The second question helps gauge how often a developer needs to debug his or her own code. Developers that need to spend a vast amount of their development time debugging may be ones who need extra help improving the code that they write.

[Read More Answers.](#)

Question # 4

Tell us do you have experience with a certain coding language/technology we use here at XYZ Company?

Answer:-

Be honest. If you don't have much experience in that particular language, tell them so - it's not like you can wing it if you make it through to the technical interview. But point out what languages you are familiar with and express that you're more than capable of learning. Remember, the desire and ability to learn is a hugely positive attribute and skill in itself.

[Read More Answers.](#)

Question # 5

Tell us if so, how big was your team and what projects did you work on?

Answer:-

Here's your chance to expand on the answer above. Make those transferrable skills shine and relate them back to the role you're applying for. Leading a team on a survival weekend to build a raft and cross a river uses the same leadership skills as leading a team of engineers to build something exciting and useful out of code that solves a problem.

[Read More Answers.](#)

Question # 6

Tell me what do you love about your current job and work environment?

Answer:-

Answer this in a balanced way. Avoid "Nothing. I hate it. The place sucks" responses. Pick a couple of elements that your company does well. The way they foster a good work/life balance or a family-like culture. Perhaps you enjoy your work, being able to create something that solves problems/fulfills a need. If this is your first job, talk about what you enjoyed while learning or interning.

[Read More Answers.](#)



Question # 7

Please explain a decision you made based on internal or external customer data?

Answer:-

Data is increasingly important. And employers like to know that you have the necessary skills to analyze data that's presented to you, draw conclusions, and make informed decisions based on your analysis. Tell the panel about a scenario where you received data, whether that's from surveys, interviews, beta testing, or customer usage, and the decision you made based on that. Did you work to improve accessibility? UX? UI? And so on. Be clear, concise, and detailed.

[Read More Answers.](#)

Question # 8

Tell me what do you know about ORM?

Answer:-

Object-relational mapping (ORM) is a way to use software code so it can map to database tables. This technique turns tables into their own classes, so then developers can use those classes for LINQ queries. The candidate might mention Hibernate, which is one of the most common ORM frameworks.

[Read More Answers.](#)

Question # 9

Please tell us did you run into any obstacles with your project and how did you handle the issue?

Answer:-

This is an important question, because it identifies how the candidate deals with obstacles, delays, and any problems that come up during the project. Most software development projects have hurdles of some kind, so a developer that has troubles identifying an obstacle with their previous project may raise a potential red flag. If you decide to hire a software developer, you need to know how they are able to overcome problems to get the job done and within the deadline.

[Read More Answers.](#)

Question # 10

Tell me what is most important when reviewing another team member's code?

Answer:-

While there's no definitive right or wrong, this question is designed to test your knowledge, how well you articulate your process, and whether you're a victim of tunnel vision. Think about it: if you say "Design. Or how well the team member's code fits into the existing architecture", you're ignoring other elements that may be of equal import to the end result. What about:

- * Functionality
- * Readability
- * Maintainability
- * Security
- * Simplicity
- * Regulatory requirements
- * Resource optimization

[Read More Answers.](#)

Question # 11

Tell us what is the most important part of your work environment that will ensure you're successful?

Answer:-

Come up with something that aligns with the culture of the company you're interviewing for. What do you need to succeed? A steady supply of coffee? A relaxed atmosphere? A quiet space to complete your work or catch your breath? A communal/collaborative workspace?

[Read More Answers.](#)

Question # 12

Explain what are your hobbies outside of work?

Answer:-

Companies like to see that you're a balanced person. Give them a little list of the things you enjoy most outside of work. Avoid saying that you're a power gamer who spends 10 hours coding at the office, rushes home, and spends another 8 hours glued to your desktop monitor decimating your virtual foes. Instead, give a sense of balance, include a few different hobbies that include individuality - like reading (pick a genre), craft and DIY type hobbies, and other solitary pursuits, as well as more social or team activities, like gaming, maybe you're a member of a field hockey club, or you are part of a local hiking group. Show the interview panel diversity in your interests. What you do outside of work tells your employers a lot about your personality and your skills, and how well you'd fit into their team. Think individuality, teamwork/collaboration, creativity, and problem-solving. Whatever you do, don't say "I don't have any hobbies."

[Read More Answers.](#)

Question # 13

Can you explain me one of your previous projects and explain how you completed it successfully?

Answer:-

This question allows the engineer to openly discuss their previous project without getting too technical. There are several advantages with this discussion. The first one is that you can identify if the candidate worked with a team and if he or she was able to work together to brainstorm solutions. It also helps you understand their management skills, time estimates, interactions with project managers, and their contribution to the overall project.

[Read More Answers.](#)

Question # 14

Please explain how do you organize your class modules and assets?



Answer:-

This will help you understand how the candidate organizes their code. There is no one "right" way to answer this question, but your team probably has a set standard and it helps to know if the developer organizes code in a way that is easily maintained and can be further documented.

[Read More Answers.](#)

Question # 15

Please explain us what projects have you been focused on at your current job?

Answer:-

Be concise. Don't ramble on and bore the recruiter. Be direct and succinctly describe the most notable/relevant projects you've been involved with at your current gig. If it's your first job, again, be honest and instead describe some of your portfolio projects.

[Read More Answers.](#)

Question # 16

Tell me what project are you most proud of in your career thus far? Why?

Answer:-

This is a time for you to shine, to tell the panel of your coding prowess, and describe the project you are most proud of. It doesn't have to be that time when you built Facebook single-handedly in a single afternoon. It can be something as simple as building an app using a new language or your work on a collaborative project that solved a real problem faced by thousands of everyday folks. Don't just brag. Yes, be confident and proud of your achievement, but try to relate the "why" to other people. "This is the project I'm most proud of because it filled a genuine need."

[Read More Answers.](#)

Question # 17

Explain me what are your thoughts on unit testing?

Answer:-

Unit testing and test driven development (TDD) are often regarded as best practices in software development and code maintenance. Unit testing is an extra set of code that tests various methods and procedures for logic errors and coding flaws. This helps eliminate many of the bugs that could be promoted to production. Every developer should be familiar with unit testing, and it's important to know how much a software developer prioritizes it in their workflow.

[Read More Answers.](#)

Question # 18

Project Management Based Senior Software Developer Job Interview Questions:

Answer:-

- * How many of the three variables scope, time and cost can be fixed by the customer?
- * Who should make estimates for the effort of a project? Who is allowed to set the deadline?
- * Do you prefer minimization of the number of releases or minimization of the amount of work-in-progress?
- * Which kind of diagrams do you use to track progress in a project?
- * What is the difference between an iteration and an increment?
- * Can you explain the practice of risk management? How should risks be managed?
- * Do you prefer a work breakdown structure or a rolling wave planning?
- * What do you need to be able to determine if a project is on time and within budget?
- * Can you name some differences between DSDM, Prince2 and Scrum?
- * How do you agree on scope and time with the customer, when the customer wants too much?

[Read More Answers.](#)

Question # 19

Testing Based Senior Software Developer Job Interview Questions:

Answer:-

- * Do you know what a regression test is? How do you verify that new changes have not broken existing features?
- * How can you implement unit testing when there are dependencies between a business layer and a data layer?
- * Which tools are essential to you for testing the quality of your code?
- * What types of problems have you encountered most often in your products after deployment?
- * Do you know what code coverage is? What types of code coverage are there?
- * Do you know the difference between functional testing and exploratory testing? How would you test a web site?
- * What is the difference between a test suite, a test case and a test plan? How would you organize testing?
- * What kind of tests would you include for a smoke test of an ecommerce web site?
- * What can you do reduce the chance that a customer finds things that he doesn't like during acceptance testing?
- * Can you tell me something that you have learned about testing and quality assurance in the last year?

[Read More Answers.](#)

Question # 20

Construction Based Senior Software Developer Job Interview Questions:

Answer:-

- * How do you make sure that your code can handle different kinds of error situations?
- * Can you explain what Test-Driven Development is? Can you name some principles of Extreme Programming?
- * What do you care about most when reviewing somebody else's code?
- * When do you use an abstract class and when do you use an interface?
- * Apart from the IDE, which other favorite tools do you use that you think are essential to you?
- * How do you make sure that your code is both safe and fast?



- * When do you use polymorphism and when do you use delegates?
- * When would you use a class with static members and when would you use a Singleton class?
- * Can you name examples of anticipating changing requirements in your code?
- * Can you describe the process you use for writing a piece of code, from requirements to delivery?

[Read More Answers.](#)

Question # 21

Requirements Based Senior Software Developer Interview Questions:

Answer:-

- * Can you name a number of non-functional (or quality) requirements?
- * What is your advice when a customer wants high performance, high usability and high security?
- * Can you name a number of different techniques for specifying requirements? What works best in which case?
- * What is requirements tracing? What is backward tracing vs. forward tracing?
- * Which tools do you like to use for keeping track of requirements?
- * How do you treat changing requirements? Are they good or bad? Why?
- * How do you search and find requirements? What are possible sources?
- * How do you prioritize requirements? Do you know different techniques?
- * Can you name the responsibilities of the user, the customer and the developer in the requirements process?
- * What do you do with requirements that are incomplete or incomprehensible?

[Read More Answers.](#)

Question # 22

Professional Senior Software Developer Job Interview Questions:

Answer:-

- * What is something substantive that you've done to improve as a developer in your career?
- * Would you call yourself a craftsman (craftsperson) and what does that word mean to you?
- * Implement a <basic data structure> using <some language> on <paper|whiteboard|notepad>.
- * What is SOLID?
- * Why is the Single Responsibility Principle important?
- * What is Inversion of Control? How does that relate to dependency injection?
- * How does a 3 tier application differ from a 2 tier one?
- * Why are interfaces important?
- * What is the Repository pattern? The Factory Pattern? Why are patterns important?
- * What are some examples of anti-patterns?
- * Who are the Gang of Four? Why should you care?
- * How do the MVP, MVC, and MVVM patterns relate? When are they appropriate?
- * Explain the concept of Separation of Concerns and it's pros and cons.
- * Name three primary attributes of object-oriented design. Describe what they mean and why they're important.
- * Describe a pattern that is NOT the Factory Pattern? How is it used and when?
- * You have just been put in charge of a legacy code project with maintainability problems. What kind of things would you look to improve to get the project on a stable footing?
- * Show me a portfolio of all the applications you worked on, and tell me how you contributed to design them.
- * What are some alternate ways to store data other than a relational database? Why would you do that, and what are the trade-offs?
- * Explain the concept of convention over configuration, and talk about an example of convention over configuration you have seen in the wild.
- * Explain the differences between stateless and stateful systems, and impacts of state on parallelism.
- * Discuss the differences between Mocks and Stubs/Fakes and where you might use them (answers aren't that important here, just the discussion that would ensue).
- * Discuss the concept of YAGNI and explain something you did recently that adhered to this practice.
- * Explain what is meant by a sandbox, why you would use one, and identify examples of sandboxes in the wild.
- * Concurrency
- * What's the difference between Locking and Lockless (Optimistic and Pessimistic) concurrency models?
- * What kinds of problems can you hit with locking model? And a lockless model?
- * What trade offs do you have for resource contention?
- * How might a task-based model differ from a threaded model?
- * What's the difference between asynchrony and concurrency?
- * Are you still writing code? Do you love it?
- * You've just been assigned to a project in a new technology how would you get started?
- * How does the addition of Service Orientation change systems? When is it appropriate to use?
- * What do you do to stay abreast of the latest technologies and tools?
- * What is the difference between "set" logic, and "procedural" logic. When would you use each one and why?
- * What Source Control systems have you worked with?
- * What is Continuous Integration? Have you used it and why is it important?
- * Describe a software development life cycle that you've managed.
- * How do you react to people criticizing your code/documents?
- * Whose blogs or podcasts do you follow? Do you blog or podcast?
- * Tell me about some of your hobby projects that you've written in your off time.
- * What is the last programming book you read?
- * Describe, in as much detail as you think is relevant, as deeply as you can, what happens when I type "cnn.com" into a browser and press "Go".
- * Describe the structure and contents of a design document, or a set of design documents, for a multi-tiered web application.
- * What's so great about <cool web technology of the day>?
- * How can you stop your DBA from making off with a list of your users' passwords?
- * What do you do when you get stuck with a problem you can't solve?
- * If your database was under a lot of strain, what are the first few things you might consider to speed it up?
- * What is SQL injection?
- * What's the difference between unit test and integration test?
- * Tell me about 3 times you failed.
- * What is Refactoring? Have you used it and it is important? Name three common refactorings.
- * You have two computers, and you want to get data from one to the other. How could you do it?
- * Left to your own devices, what would you create?



- * Given Time, Cost, Client satisfaction and Best Practices, how will you prioritize them for a project you are working on? Explain why.
- * What's the difference between a web server, web farm and web garden? How would your web application need to change for each?
- * What value do daily builds, automated testing, and peer reviews add to a project? What disadvantages are there?
- * What elements of OO design are most prone to abuse? How would you mitigate that?
- * When do you know your code is ready for production?
- * What's YAGNI? Is this list of questions an example?
- * Describe to me some bad code you've read or inherited lately.

[Read More Answers.](#)

Question # 23

Data Structures Based Senior Software Developer Job Interview Questions:

Answer:-

- * How would you implement the structure of the London underground in a computer's memory?
- * How would you store the value of a color in a database, as efficiently as possible?
- * What is the difference between a queue and a stack?
- * What is the difference between storing data on the heap vs. on the stack?
- * How would you store a vector in N dimensions in a datatable?
- * What type of language do you prefer for writing complex data structures?
- * What is the number 21 in binary format? And in hex?
- * What is the last thing you learned about data structures from a book, magazine or web site?
- * How would you store the results of a soccer/football competition (with teams and scores) in an XML document?
- * Can you name some different text file formats for storing unicode characters?

[Read More Answers.](#)

Question # 24

Functional Design Based Senior Software Developer Job Interview Questions:

Answer:-

- * What are metaphors used for in functional design? Can you name some successful examples?
- * How can you reduce the user's perception of waiting when some functions take a lot of time?
- * Which controls would you use when a user must select multiple items from a big list, in a minimal amount of space?
- * Can you name different measures to guarantee correctness of data entry?
- * Can you name different techniques for prototyping an application?
- * Can you name examples of how an application can anticipate user behavior?
- * Can you name different ways of designing access to a large and complex list of features?
- * How would you design editing twenty fields for a list of 10 items? And editing 3 fields for a list of 1000 items?
- * What is the problem of using different colors when highlighting pieces of a text?
- * Can you name some limitations of a web environment vs. a Windows environment?

[Read More Answers.](#)

Question # 25

Technical Design Based Senior Software Developer Job Interview Questions:

Answer:-

- * What do low coupling and high cohesion mean? What does the principle of encapsulation mean?
- * How do you manage conflicts in a web application when different people are editing the same data?
- * Do you know about design patterns? Which design patterns have you used, and in what situations?
- * Do you know what a stateless business layer is? Where do long-running transactions fit into that picture?
- * What kinds of diagrams have you used in designing parts of an architecture, or a technical design?
- * Can you name the different tiers and responsibilities in an N-tier architecture?
- * Can you name different measures to guarantee correctness and robustness of data in an architecture?
- * Can you name any differences between object-oriented design and component-based design?
- * How would you model user authorization, user profiles and permissions in a database?
- * How would you model the animal kingdom (with species and their behavior) as a class system?

[Read More Answers.](#)

Question # 26

Configuration Management Based Senior Software Developer Job Interview Questions:

Answer:-

- * Do you know what a baseline is in configuration management? How do you freeze an important moment in a project?
- * Which items do you normally place under version control?
- * How can you make sure that team members know who changed what in a software project?
- * Do you know the differences between tags and branches? When do you use which?
- * How would you manage changes to technical documentation, like the architecture of a product?
- * Which tools do you need to manage the state of all digital information in a project? Which tools do you like best?
- * How do you deal with changes that a customer wants in a released product?
- * Are there differences in managing versions and releases?
- * What is the difference between managing changes in text files vs. managing changes in binary files?
- * How would you treat simultaneous development of multiple RfC's or increments and maintenance issues?

[Read More Answers.](#)

Question # 27

Tell me do you have any leadership experience?



Answer:-

Don't be fooled into thinking this question relates solely to leadership in tech. If you've lead a team of remote freelance designers, took a leadership role on a volunteering experience, or even during school, say so and make it clear that those skills you gained elsewhere are absolutely transferable. Don't forget to mention that even in a leadership role, you still function as a part of your team, because collaboration and team work are vital.

[Read More Answers.](#)

Question # 28

Tell us how do you prepare your code for different kinds of errors?

Answer:-

Self-explanatory. The interviewer wants you to show your knowledge and give a reasonably polished answer that tells them you aren't just "winging" it.

[Read More Answers.](#)

Question # 29

Tell me do you know what is MVC?

Answer:-

MVC stands for Model-View-Controller. It's a way to separate code into its own compartments, typically like so:

- * Model is the data layer that manages business logic and maintains data
- * View is the front-end presentation layer, or UI layer
- * Controller is the logic layer that handles user requests and behavior, and updates the Model

[Read More Answers.](#)

Question # 30

Maintenance Based Senior Software Developer Job Interview Questions:

Answer:-

- * What kind of tools are important to you for monitoring a product during maintenance?
- * What is important when updating a product that is in production and is being used?
- * How do you find an error in a large file with code that you cannot step through?
- * How can you make sure that changes in code will not affect any other parts of the product?
- * How do you create technical documentation for your products?
- * What measures have you taken to make your software products more easily maintainable?
- * How can you debug a system in a production environment, while it is being used?
- * Do you know what load balancing is? Can you name different types of load balancing?
- * Can you name reasons why maintenance of software is the biggest/most expensive part of an application's life cycle?
- * What is the difference between re-engineering and reverse engineering?

[Read More Answers.](#)

Question # 31

Algorithms Based Senior Software Developer Job Interview Questions:

Answer:-

- * How do you find out if a number is a power of 2? And how do you know if it is an odd number?
- * How do you find the middle item in a linked list?
- * How would you change the format of all the phone numbers in 10,000 static html web pages?
- * Can you name an example of a recursive solution that you created?
- * Which is faster: finding an item in a hashtable or in a sorted list?
- * What is the last thing you learned about algorithms from a book, magazine or web site?
- * How would you write a function to reverse a string? And can you do that without a temporary string?
- * What type of language do you prefer for writing complex algorithms?
- * In an array with integers between 1 and 1,000,000 one value is in the array twice. How do you determine which one?
- * Do you know about the Traveling Salesman Problem?

[Read More Answers.](#)

Question # 32

Explain me a project that failed. What happened? How did you rebound?

Answer:-

Use this question to tell the interviewers that you are able to clearly articulate why a project didn't work. You took the time to dissect the project and pinpoint what went wrong. Then you used that knowledge to better yourself, moving on to new projects and ensuring you didn't repeat those same steps that resulted in the project failure. The panel wants to know you recognize that you're human, that you're aware that you're not infallible, and that you own your errors. They also want to see how you move on from that.

[Read More Answers.](#)

Question # 33

Please explain your coding process from start to finish?

Answer:-

Right from the start. What's the first thing you do when you get an assignment? Get a pen and paper and break it down into small chunks/tasks/milestones? List the start and end dates in your calendar? Make sure you prepare for this question. Employers use this to make sure you have a clear process and are organized rather than haphazard and chaotic in your approach to work. Walk through your process in preparation for the interview so you can clearly articulate how you work.

[Read More Answers.](#)



Question # 34

Tell us which tools do you like to use for keeping track of requirements?

Answer:-

Have an answer ready for this one. And be honest. You don't have to be all high-tech all of the time. It's okay to say that you keep track of daily tasks with pen and paper. Maybe you blend old-school paper lists with Trello boards, Asana, or some other project management app. Just go ahead and list what you use. But don't say "I just remember them"!

[Read More Answers.](#)

Question # 35

Tell me what excites you about joining our team here?

Answer:-

Just like when you talked to the recruiter, talk knowledgeably about what elements of the company attract you the most to show you've done your research, but don't be sycophantic or show a creepy stalker-esque level of "I love you/your company soooooo much because...".

[Read More Answers.](#)

Question # 36

Explain me your ideal company culture?

Answer:-

Do your research on this one. Don't tell them you love nothing more than silence and rigid discipline if their company culture is relaxed and fun. Make sure what you tell them roughly aligns with what you have learned about their company.

[Read More Answers.](#)

Question # 37

Explain me how do you design scalable applications?

Answer:-

Again, this is self-explanatory and designed to test your knowledge and thought processes. What factors do you look at and plan for to ensure you avoid performance bottlenecks? Is the interviewer talking about scaling up, scaling out, or both?

[Read More Answers.](#)

Question # 38

Tell me do you follow the Agile software development process?

Answer:-

Today, Agile is the most common way to manage software development procedures. Software developers should be on board and familiar with the process, so you can ask them if they are willing to work under those circumstances.

[Read More Answers.](#)

Question # 39

Tell me how familiar are you with OOP analysis and design?

Answer:-

Object-oriented programming (OOP) is the core of major languages such as VB, Java, C++, and C#, so top software developers should have strong OOP skills. OOP is a process of using classes to describe components of a program. For instance, if your program explains a house, classes would be the roof, doors, windows, and rooms. Your programmer should be able to explain classes using a similar analogy.

[Read More Answers.](#)

Question # 40

Please explain how would co-workers describe the role you play on the team?

Answer:-

Use this question to show the interviewers that you are socially aware - that you are perceptive enough to recognize how you are perceived by others, particularly your peers. Also use this question to show that you're self-aware. You know your own strengths and weaknesses, and you know how others see you and what you bring to the team. Be honest without being overly self-deprecating.

[Read More Answers.](#)

Question # 41

Tell us what is your preferred language for complex algorithms?

Answer:-

Be honest but try to give at least two answers. Again, giving more than one response shows you are agile and versatile and not stuck on one thing. "XYZ is my first choice, but ABC is also a great option." Then tell them why.

[Read More Answers.](#)

Question # 42

Explain me what do you know about multi-threading?

Answer:-

Multi-threading is a way to improve the performance of an application. Multi-threading is a process that uses several processors to run snippets of code. It improves



performance and stops code from "hanging" when it runs.

[Read More Answers.](#)

Global Guideline - COM

Computer Programming Most Popular Interview Topics.

- 1 : [PHP Frequently Asked Interview Questions and Answers Guide.](#)
- 2 : [C++ Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 3 : [C Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 4 : [Software engineering Frequently Asked Interview Questions and Answers Guide.](#)
- 5 : [Cobol Frequently Asked Interview Questions and Answers Guide.](#)
- 6 : [Visual Basic \(VB\) Frequently Asked Interview Questions and Answers Guide.](#)
- 7 : [Socket Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 8 : [Perl Programming Frequently Asked Interview Questions and Answers Guide.](#)
- 9 : [VBA Frequently Asked Interview Questions and Answers Guide.](#)
- 10 : [OOP Frequently Asked Interview Questions and Answers Guide.](#)

About Global Guideline.

Global Guideline is a platform to develop your own skills with thousands of job interview questions and web tutorials for fresher's and experienced candidates. These interview questions and web tutorials will help you strengthen your technical skills, prepare for the interviews and quickly revise the concepts. Global Guideline invite you to unlock your potentials with thousands of [Interview Questions with Answers](#) or begin a tutorial right away, such as [HTML](#), [XML](#), [XSLT](#), [Cascading Style Sheet \(CSS\)](#), [Search Engine Optimization \(SEO\)](#), [JavaScript](#), [Structure Query Language \(SQL\)](#), [Database Articles](#), [Web Hosting Guide](#) and much more. Learn the most common technologies [Interview Questions and Answers](#). We will help you to explore the resources of the World Wide Web and develop your own skills from the basics to the advanced. Here you will learn anything quite easily and you will really enjoy while learning. Global Guideline will help you to become a professional and Expert, well prepared for the future.

* This PDF was generated from <https://www.GlobalGuideline.com> at **April 4th, 2021**

* If any answer or question is incorrect or inappropriate or you have correct answer or you found any problem in this document then don't hesitate feel free and [e-mail us](#) we will fix it.

You can follow us on FaceBook for latest Jobs, Updates and other interviews material.
www.facebook.com/InterviewQuestionsAnswers

Follow us on Twitter for latest Jobs and interview preparation guides
<https://twitter.com/InterviewGuide>

Best Of Luck.

Global Guideline Team
<https://www.globalguideline.com>
Support@globalguideline.com