

1. What are your technical certifications?
 - Certified data professional (CDP)
 - Certified information systems security professional (CISSP)
 - Cisco certified internetwork expert (CCIE)
 - Cisco certified network associate (CCNA)
 - Cisco certified network professional (CCNP)
 - Certified information systems auditor (CISA)
 - CompTIA A+
 - Microsoft technology associate (MTA)
 - Project management professional (PMP)
 - Oracle certified professional
 - Salesforce certified development lifecycle and deployment
 - Certified Scrum master (CSM)
 - AWS certified solutions architect
 - Certified ethical hacker (CEH)
 - Global information assurance certification (GIAC)
 - ITIL
2. How did your education prepare you for this job?
 - a. My education provided me with the coding skills I needed to be successful as a Software Engineer. My minor in Mathematics allows me to take my programming skills to the next level.
3. What do you do to maintain your technical certifications?
 - a. Take professional development courses
 - b. Utilize online resources
 - c. Attend professional events
 - d. Network online
 - e. Invest in continuing education and certifications
4. How would you rate your key competencies for this job?
 - a. Through my education, work experience and various projects I worked on, I have had extensive experience and exposure to approximately 12 different programming languages and multiple operating systems. I am most adept at Java, C, C#, C++ and SQL.
5. What are your IT strengths and weaknesses?
 - a. My strengths would be my extensive experience and education in software design, application development and object-oriented programming principles. I also possess excellent organization and interpersonal skills. I am a fast learner with solid time management and multi-tasking skills, and a strong, collaborative work ethic. I have excellent troubleshooting, problem-solving and verbal and written communications

skills and I have a systematic, organized, analytical mindset. I have been successful in team and individual settings to drive product success and process efficiency.

- b. My weaknesses would be that I sometimes lack self-confidence could use more experience in encryption and cryptography.
6. Tell me about the most recent project you worked on. What were your responsibilities?
7. Describe a time you were able to improve upon the design that was originally suggested.
8. Tell me about the project you are most proud of, and what your contribution was.
9. Describe your production deployment process.
 - a. The planning stage is the most important as this is where you define the timelines, delivery dates, requirements and the overall scope of the project.
 - b. Next, you build the release
 - c. Next is the user acceptance testing (this is an iterative process as the build will go back and forth as issues are identified, fixed and retested
 - d. Once the testing is completed, you prepare for the release. In this part of the process, you should prepare for any issues that come up at deployment.
 - e. Deploy release
10. Give an example of where you have applied your technical knowledge in a practical way.
11. How did you manage source code?
 - a. Source Code Management primarily involves tracking the modifications to code. Tracking modifications assists development and collaboration by providing a running history of development and helping to resolve conflicts when merging contributions from multiple sources. Software tools for source code management are sometimes referred to as "Source Code Management Systems" (SCMS), "Version Control Systems" (VCS), "Revision Control Systems" (RCS) or simply "code repositories" depending on what features they provide or how they are being used. A "repository" typically refers to one source code development project within a system.
12. What did you do to ensure quality in your deliverables?
 - a. Internal and/or external quality assurance
 - b. Review and feedback throughout the project
 - c. Create and follow a QA checklist
13. When is the last time you downloaded a utility from the internet to make your work more productive, and what was it?

- a. AutoHotKey. This software can automate just about anything by capturing your keystrokes and mouse clicks. This free utility allows you to define your own hotkeys, enabling you to launch an application with a single key press. Fortunately, this application does come with a built-in macro recorder.
14. From the description of this position, what do you think you will be doing on a day-to-day basis?
- a. Planning, conducting, and coordinating software development activities. Design, develop, document, test, and debug software that contains logical and mathematical solutions to business/mission problems or questions in computer language for solutions by means of data processing equipment. Involved in the entire development life cycle. Apply your computer hardware and software knowledge for programming initiatives. Correcting program errors, preparing operating instructions, compile documentation of program development, and analyze system capabilities to resolve questions.
15. What challenges do you think you might expect in this job if you were hired?
- a. It is usually best to choose aspects of the job that aren't absolutely critical to your specific role. For example, if you're a journalist applying for a position as a web editor, you might mention you're working on improving your photojournalism or video skills. Don't say that proofreading or writing copy will be your biggest challenge because as a web editor these are core responsibilities of the position.

Selecting a knowledge or skill area that you lack is generally more advisable than choosing a personality trait that would be hard to change. For example, if you're applying for a sales position, you would not want to mention that reaching out to new people makes you nervous.

Instead, you might mention that you have modest skills in PowerPoint but would be glad to take workshops or complete online tutorials to upgrade your skills.
16. What is the difference between functional requirements and non-functional requirements?
- a. Functional requirements are the features that a developed software product is expected to perform. For example, adding a payment option at an eCommerce website will be a functional requirement. Non-functional requirements measure the usability of the application such as User Interface look and feel, Security, Performance, Interoperability, Reliability, etc.

17. You have been asked to research a new business tool. You have come across two solutions. One is an on-premises solution, the other is cloud-based. Assuming they are functionally equivalent, would you recommend one over the other, and why?

A hybrid cloud solution is a solution that features an element of different types of IT deployment models, ranging from on premises to private cloud and public cloud. A hybrid cloud infrastructure depends on the availability of a public cloud platform from a trusted third-party provider, a private cloud constructed either on premises or through a hosted private cloud provider, and effective WAN connectivity between both of those environments.

a. Deployment

On Premises: In an on-premises environment, resources are deployed in-house and within an enterprise's IT infrastructure. An enterprise is responsible for maintaining the solution and all its related processes.

Cloud: While there are different forms of cloud computing (such as public cloud, private cloud, and a hybrid cloud), in a public cloud computing environment, resources are hosted on the premises of the service provider but enterprises are able to access those resources and use as much as they want at any given time.

b. Cost

On Premises: For enterprises that deploy software on premise, they are responsible for the ongoing costs of the server hardware, power consumption, and space.

Cloud: Enterprises that elect to use a cloud computing model only need to pay for the resources that they use, with none of the maintenance and upkeep costs, and the price adjusts up or down depending on how much is consumed.

c. Control

On Premises: In an on-premises environment, enterprises retain all their data and are fully in control of what happens to it, for better or worse. Companies in highly regulated industries with extra privacy concerns are more likely to hesitate to leap into the cloud before others because of this reason.

Cloud: In a cloud computing environment, the question of ownership of data is one that many companies – and vendors for that matter, have struggled with. Data and encryption keys reside within your third-party provider, so if the unexpected happens and there is downtime, you maybe be unable to access that data.

d. Security

On Premises: Companies that have extra sensitive information, such as government and [banking industries](#) must have a certain level of security and privacy that an on-premises environment provides. Despite the promise of the cloud, security is the primary concern for many industries, so an on-premises environment, despite some of its drawbacks and price tag, make more sense.

Cloud: Security concerns remain the number one barrier to a cloud computing deployment. There have been many publicized cloud breaches, and IT departments around the world are concerned. From personal information of employees such as login credentials to a loss of intellectual property, the security threats are real.

e. Compliance

On Premises: Many companies these days operate under some form of [regulatory control](#), regardless of the industry. Perhaps the most common one is the Health Insurance Portability and Accountability Act (HIPAA) for private health information, but there are many others, including the Family Educational Rights and Privacy Act (FERPA), which contains detailed student records, and other government and industry regulations. For companies that are subject to such regulations, it is imperative that they remain compliant and know where their data is at all times.

Cloud: Enterprises that do choose a cloud computing model must do their due diligence and ensure that their third-party provider is up to code and in fact compliant with all of the different regulatory mandates within their industry. Sensitive data must be secured, and customers, partners, and employees must have their privacy ensured.

18. What would you do to ensure you provided accurate project estimates?

- a. Maintain an ongoing "actual hours" database of the recorded time spent on each aspect of your projects. Use the data to help estimate future projects and identify the historically accurate buffer time needed to realistically perform the work
- b. Create and use planning documents, such as specifications and project plans
- c. Perform a detailed task analysis of the work to be performed
- d. Use a "complexity factor" as a multiplier to determine whether a pending project is more or less complex than a previous one
- e. Use more than one method to arrive at an estimate, and look for a midpoint among all of them

- f. Identify a set of caveats, constraints, and assumptions to accompany your calculations, which would bound the conditions under which your estimates would be meaningful. (Anything that occurs outside of those constraints would be considered out of scope.)
 - g. If the proposed budget or schedule seems inadequate to do the work, propose adjusting upward or downward one or more of the four project scoping criteria: cost, schedule, quality, and features
 - h. Consider simpler or more efficient ways to organize and perform the work
 - i. Plan and estimate the project rollout from the very beginning so that the rollout won't become a chaotic scramble at the end. For instance, you could propose using a minimally disruptive approach, such as a pilot program or a phased implementation
 - j. In really nebulous situations, consider a phase-based approach, where the first phase focuses primarily on requirements gathering and estimating
 - k. Develop contingency plans by prioritizing the deliverables right from the start into "must-have" and "nice-to-have" categories
 - l. Refer to your lessons-learned database for "20:20 foresight" on new projects, and incorporate your best practices into future estimates
- 19. You have learned that a business unit is managing a major component of the business using Excel spreadsheets and Access databases. What risks does this present, and what would you recommend be done to mitigate those risks?
- 20. What development tools have you used?
 - a. Monday.com
 - b. Embold
 - c. Linx
 - d. AWS Cloud9
 - e. GitHub
 - f. NetBeans
 - g. Bootstrap
- 21. What languages have you programmed in?
 - a. Java
 - b. C#

- c. C
- d. C++
- e. PHP
- f. Python
- g. JavaScript
- h. Dart

22. What source control tools have you used?

- a. Git
- b. Perforce Helix Core
- c. Subversion
- d. ClearCase
- e. Team Foundation Server
- f. Mercurial
- g. CVS

23. Describe the elements of an n-tier architecture and their appropriate use.

- a. An **N-Tier Application** program is one that is distributed among three or more separate computers in a distributed network.

The most common form of n-tier is the 3-tier Application, and it is classified into three categories.

- User interface programming in the user's computer
- Business logic in a more centralized computer, and
- Required data in a computer that manages a database.

This architecture model provides Software Developers to create Reusable application/systems with maximum flexibility.

24. Compare and contrast REST and SOAP web services.

- a. **SOAP** stands for Simple Object Access Protocol whereas **REST** stands for Representational State Transfer. **SOAP** is a protocol whereas **REST** is an architectural

pattern. ... **SOAP** only works with XML formats whereas **REST** work with plain text, XML, HTML and JSON. **SOAP** cannot make use of **REST** whereas **REST** can make use of **SOAP**

25. Define authentication and authorization and the tools that are used to support them in enterprise deployments.

Authentication	Authorization
Authentication confirms your identity to grant access to the system.	Authorization determines whether you are authorized to access the resources.
It is the process of validating user credentials to gain user access.	It is the process of verifying whether access is allowed or not.
It determines whether user is what he claims to be.	It determines what user can and cannot access.
Authentication usually requires a username and a password.	Authentication factors required for authorization may vary, depending on the security level.
Authentication is the first step of authorization so always comes first.	Authorization is done after successful authentication.
For example, students of a particular university are required to authenticate themselves before accessing the student link of the university's official website. This is called authentication.	For example, authorization determines exactly what information the students are authorized to access on the university website after successful authentication.

26. Have you used Visual Studio?
27. Have you used Eclipse?
28. What is a SAN, and how is it used?
- A storage area network (**SAN**) or storage network is a computer network which provides access to consolidated, block-level data storage. SANs are primarily **used** to access storage devices, such as disk arrays and tape libraries from servers so that the devices appear to the operating system as direct-attached storage.
29. What is clustering, and describe its use?
- Cluster** analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (**clusters**). ... **Clustering** can therefore be formulated as a multi-objective optimization problem.
30. What is the role of the DMZ in network architecture?
- The goal of a **DMZ** is to add an extra layer of security to an organization's local area **network**. A protected and monitored **network** node that faces outside the internal **network** can access what is exposed in the **DMZ**, while the rest of the organization's **network** is safe behind a firewall.

31. What is a cross-site scripting attack, and how do you defend against it?
- Cross-site Scripting (XSS) is a client-side code **injection attack**. The attacker aims to execute malicious scripts in a web browser of the victim by including malicious code in a legitimate web page or web application. The actual attack occurs when the victim visits the web page or web application that executes the malicious code. The web page or web application becomes a vehicle to deliver the malicious script to the user's browser.
 - To protect against Cross-site Scripting, you must scan your website or web application regularly or at least after every change in the code. Then, your developers must correct the code to eliminate the vulnerability. Contrary to popular opinions, web application firewalls do not protect against Cross-site Scripting, they just make the attack more difficult – the vulnerability is still there.
32. In network security, what is a honeypot, and why is it used?
- A **honeypot** is a computer or computer system intended to mimic likely targets of cyberattacks. It can be **used** to detect attacks or deflect them from a legitimate target. It can also be **used** to gain information about how cybercriminals operate.
33. Tell me about some of the databases you have designed.
34. How do you enforce relational integrity in database design?
35. When is it appropriate to de-normalize database design?
36. What is the difference between OLAP and OLTP? When is each used?
37. What automated-build tools or processes have you used?
38. What is the role of continuous integration systems in the automated-build process?
39. Describe the difference between optimistic and pessimistic locking.
40. In databases, what is the difference between a delete statement and a truncate statement?
41. What are transaction logs, and how are they used?
42. What are the most important database performance metrics, and how do you monitor them?
43. What is the role of SNMP?
44. How important is it to work directly with your business users?
45. What elements are necessary for a successful team and why?
46. What percentage of your time do you spend unit testing?
47. What do you expect in the solution documents you are provided?
48. How much reuse do you get out of the code that you develop, and how?
49. Which do you prefer; service-oriented or batch-oriented solutions?

50. What technical websites do you follow?
51. What programming languages have you used?
- a. A software engineer should have experience with a wide range of programming languages. This knowledge is vital to having success in this role. Before your interview, review the job description to see if they mention whether or not you'll need programming language experience for the role. Respond with languages you're familiar with that match the employer's needs.
- Example:** *"I am proficient in Java, C++, JavaScript, C#, Ruby and Python. Of these programming languages, I feel most comfortable working with Java, C# and C++. In my previous role, I worked mainly with Java to create applications that worked across multiple platforms. I also used C++ to develop a new operating system that worked with the applications I engineered. Using C#, I was able to improve my productivity when developing web-based apps and software."*
52. Describe the process you use for writing a piece of code, from requirements to delivery.
53. What books have you read on software engineering that you would recommend to someone in the business?
54. How do you make sure that your code can handle different kinds of error situations?
55. How do you find an error in a large file with code that you cannot step through?
56. How do you design scalable applications? Walk us through your process.
57. What is the difference between a mutex and a semaphore? Which would you use to protect access to an increment operation?
- a. Both mutexes and semaphores are used to control access to a shared resource – most often in multithreading scenarios. A mutex is used when only one thread or process is allowed to access a resource and a semaphore is used when only a certain set limit of threads or processes can access the shared resource. Essentially a mutex is a semaphore where the limit is set to 1.
- Which one would I use to protect access to an increment operation? Well, if I'm using C# I would say neither – just use `Interlocked.Increment` – but in the general scenario I would use a mutex.
58. What is the difference between re-engineering and reverse engineering?
59. What is the difference between local and global variables?
60. What is the agile software philosophy?

61. Name one or two examples of how an application can anticipate user behavior.
62. What would you do if a coworker asked you to review their code, and it was full of errors?
63. Describe your ideal level of interaction with coworkers that would allow you to achieve the most success.
64. Tell me about a time you worked with coworkers to solve an issue at work.
65. Tell me about a time when you had to solve a problem, but you didn't have all the necessary information about it in hand.
66. Imagine your manager wants to buy new software for the office, but you think it will decrease productivity. What do you do?
67. What distinguishes a great software engineer from a good one? Do you feel you have those qualities?
 - a. "A great software engineer has a healthy balance between perfectionism and pragmatism. Too often engineers want their code to be perfect, while losing sight of the overall goals of the project. A great programmer also learns not to fall in love with their own code, to keep a healthy skepticism until it's been thoroughly tested, making sure it is the right choice for the project at hand. And yes, I believe have the qualities of a great programmer, though working on my need to be perfect is an ongoing battle."
68. What's the most important thing to look for or check when reviewing another team member's code?
 - a. "Some things you want to look for when reviewing a team member's code is its functionality and how readable it is. Is it secure, or are there obvious flaws that would cause security problems and make it easy to hack? Is the code simple, or are there a lot of unnecessary line of code that needs to be removed or rewritten? Does it meet the regulatory requirements in place for the project, and is it optimized to not be resource-heavy?"
69. In your opinion, what are the principles of good software engineering? What are some basic principles everyone should follow?
 - a. "I think one of the main principles of software engineering, and one I try to live by, is to keep things as simple as possible. You're often already dealing with complex algorithms and design concerns, so no need to make things even more difficult with overly complicated, resource-heavy code. Your code should be simple, lean and easy to read. If you start there, the rest will follow."
70. If needed, how would you go about designing scalable applications? Walk us through your process.

- a. "You design scalable projects by writing as little code as possible. You also want to reuse as much code as you can while not over or under doing your design. Then break things down into modular sections that can run on multiple or separate systems."

71. Tell us a bit about the latest project you worked on. Was it completed successfully? Explain how you contributed to its success and how you handled any obstacles you may have run into.

- a. A question relating to a specific project will help the interviewer better understand your process and how you handle adversity while working on a specific task.

As you answer this question, use the [STAR method](#) (Situation, Task, Action and Result) to craft a detailed, informative answer. Start by describing the situation, which allows you to provide all the necessary details about the project on which you were working. Next, discuss the tasks you were assigned to outline your level of responsibility. When describing the action you took, include the steps you took to achieve a goal, and finish with the result of the project.

Example: *"A previous employer tasked me with creating an internal online learning and training program for employees. The purpose of the program was to make sure all employees received proper training on certain topics, including customer service, compliance with legal requirements and workplace ethics. I began by researching other similar training systems to figure out what worked and what didn't. Next, I used Java to code a simple program, which I then used to upload training courses. After testing the simplified program, I added elements to make it more engaging to employees, such as games and interactive quizzes. This program was well-received by the employees of the organization, and their customer service success rates increased by 25% after all team members took the required course."*

72. What has your experience been like as part of an agile software development process, if any?

73. Here is a simple programming challenge. Could you have a go at solving it?

- a. Example question 1 (shorter time frame): Write a function to compute the Nth Fibonacci number.
- b. Example question 2 (longer time frame): Write a function that takes the current position of a knight on a chessboard, and returns a preliminary list of possible moves the knight could make. (That is, the current positions of other pieces are not provided, so you can't

check against capturing pieces on the knight's own side nor making their king vulnerable to capture.)

74. What do you think are the most important aspects to pay attention to when reviewing another team member's code?

- a. Code reviews are fundamental to the software development process, even when there's only one engineer. By posing this question you'll get an idea of the candidate's knowledge and problem-solving skills, their attention to detail, and whether they can keep an overview of the project.

Here is a sample answer:

"I first look for security, functionality, and readability. Is the code simple, or cluttered, bloated, and inefficient? How many lines of unnecessary code will I need to re-write or remove? I check for any weaknesses that could cause vulnerabilities and confirm that regulatory requirements have been met."

Everyone has their own coding style and every developer or team will have requirements that are specific to their codebase. Effective code reviews often have checklists. Below is a limited list of general suggestions you could consider including:

1. The software passes automated and manual testing
2. Code follows applicable conventions and is easy to understand
3. Code is not duplicated
4. No negatively named boolean variables
5. Scrutinize [methods with boolean parameters](#)
6. Blocks of code inside loops are as small as possible
7. No memory leaks

But more important than which exact points a candidate brings up is their reasoning for doing so. Be wary of candidates who get stuck on tabs-versus-spaces bikeshedding at the expense of more crucial engineering elements: The above items shouldn't all carry the same weight.

75. Do you consider unit testing essential or a waste of time?

- a. Every engineer/developer worth considering should be familiar with unit testing. Asking this question will give you an understanding of their attitude toward it, and what level of priority they give it in their working process. Do they follow test-driven development

(TDD) or behavior-driven development (BDD), or are unit tests something they tack on afterward for the sake of process conformance or mere appearances?

- b. Typically regarded by most industry professionals as being a best practice in code maintenance and software development, unit tests are usually part of an overall testing strategy. They test for logic errors and coding flaws, helping to prevent bugs from advancing to the finished product. Plus, because they're automated, they prevent *regressions*, where bugs return that had already been fixed.

76. What has your experience been like as part of an agile software development process, if any?

- a. The Manifesto for Agile Software Development outlines an approach based on iterations rather than a waterfall model. Requirements and solutions are generated through the collaboration of self-organizing and cross-functional teams and their end users. Among other things, it encourages a flexible planning style and a rapid response to change.
- b. Knowing how a developer feels about agile development can help you understand how they will fit into your own process. Open-minded developers that are also able to see flaws in how agile processes have been run can provide valuable feedback to help your team's methodology grow and evolve.
- c. On the other hand, if they're dead-set against a core process of yours, there may end up being too much friction for them to stay productive.

77. How familiar are you with object-oriented programming (OOP)?

- a. OOP has been a standard convention for over 20 years and is organized around objects rather than actions, and data rather than logic. It is ever-present and it is very unlikely a candidate would not have run into it at some point.

Here are ten examples of terms they should be able to define:

- i. class, object (and the difference between the two)
- ii. method (as opposed to, say, a C function)
- iii. virtual method, pure virtual method
- iv. class/static method
- v. static/class initializer
- vi. constructor
- vii. destructor/finalizer
- viii. superclass or base class
- ix. subclass or derived class

- x. NB. Some firms avoid OOP and prefer to use a functional programming (FP) language such as Clojure.

78. Please explain big-O notation in the simplest terms.

- a. Big-O notation (Landau's symbol) is used in computer science to describe the performance or complexity of an algorithm. It describes how the runtime or space requirement of a function grows as the input grows.

Two functions with the same Big-O notation will tend to have the same growth rate and thus have the same relative performance with large inputs.

For example, the bubble sort algorithm has an average time complexity of $O(n^2)$ while merge sort and heap sort both have an average complexity of $O(n \log n)$. In average cases, merge sort and heap sort will demonstrate similar performance while they will both outperform bubble sort.

Candidates should be able to demonstrate a basic understanding of the fundamentals of big-O algorithmic complexity analysis.

They should know that algorithms usually fall into the following performance classes:

- i. Constant-time
- ii. Logarithmic
- iii. Linear
- iv. Polynomial
- v. Exponential
- vi. Factorial

They should also be able to explain why a given operation falls to a particular complexity class.

79. What is SDLC OR Software Development Life Cycle?

- a. The process of planning, creating, testing and deploying a software is called Software Development Life Cycle or SDLC. Different tasks to be performed in each step of the software development process is explained well in SDLC. Different phases of SDLC are

planning, requirements, design, development, testing, deployment and maintenance.

Various SDLC models are waterfall model, spiral model, V-shaped model, iterative model, big bang model and agile model. Each part of software development is evaluated easily and helps programmers to work concurrently using SDLC phases. This is a process to be followed and not a technique.

- i. Requirement Phase
- ii. Analysis Phase
- iii. Design Phase
- iv. Development Phase
- v. Testing Phase
- vi. Deployment & Maintenance Phase

80. How familiar are you with Rhapsody?

- a. Rhapsody is part of the IBM Engineering portfolio that provides a collaborative design development, and test environment for systems engineers that supports UML, SysML, UAF and AUTOSAR. The solution also allows for control of defense frameworks (DoDAF, MODAF and UPDM) and helps accelerate industry standards such as DO-178, DO-178B/C, and ISO 26262. It provides continuous validation, automatic consistency checking and allows you to share, collaborate and review engineering lifecycle artifacts.

81. What is verification and validation?

- a. Verification: Verification is a term that refers to the set of activities which ensure that software implements a specific function.
- b. Validation: It refers to the set of activities which ensure that software that has been built according to the need of clients.

82. In software development process what is the meaning of debugging?

- a. Debugging is the process that results in the removal of error. It is very important part of the successful testing.

83. How can you make sure that your code is both safe and fast?

- a. In the software, development security is always first. So if the execution of the program is slow then, I will try to identify the reason out ways to its time complexity.

84. Name two tools which are used for keeping track of software requirements?

- a. There many I ways to keep track of requirements.

85. Two commonly used are:

- a. Make a requirements specifications document to list all of the requirements.
 - b. Create an excel sheet the list down the requirement, type, dependency, priority, etc.
86. What is the main difference between a stubs, a mock?
- a. A stub is a minimal implementation of an interface which generally returns hardcoded data while mock usually verifies outputs against expectations. Those expectations are set in the test.
87. What language do you like to write programming algorithms?
- a. Every developer has their views when it comes to the programming language choices. Though, one should prefer high-level languages because they are dynamic. Like C and C++ languages.
88. What is computer software?
- a. Computer software is a package which includes a software program, its documentation, and user guide on how to use the software.
89. According to you which SDLC model is the best?
- a. There, is no such ranking, as SDLC Models are adopted as per the need for the development process. It may differ software-to-software.
90. Who is software project manager? What is his role?
- a. A software project manager is a person responsible for managing the software development project.
 - b. The project manager is doing the project planning, monitoring the progress, communication. He or she also manages risks and resources to deliver the project within time, cost, and quality constraints.
91. What is mean by software scope?
- a. Software scope is a well-defined boundary. It includes all kind of activities that are done to develop and deliver the software product.
 - b. The software scope defines all functionalities and artifacts to be delivered as a part of the software. The scope also identifies what the product will do? What is not the part of the project? What is project estimation?
 - c. This process is helpful to estimate various aspects of the software product. This estimation can be decided either consulting experts or by using pre-defined formulas.
92. How to find the size of a software product?
- a. The size of software product can be calculated using by following two methods

- i. Counting the lines of delivered code
 - ii. Counting delivered function points
- 93. What are function points?
 - a. Function points are the features which are provided by the software product. It is considered as a most important measurement for software size.
- 94. What are software project estimation techniques available?
 - a. Most widely used estimation techniques are:
 - i. Decomposition technique
 - ii. Empirical technique
- 95. What is Software configuration management?
 - a. Software configuration management is a process of tracking and controlling changes that happen in the software.
 - b. Change control is a function which ensures that all changes made into the software system are consistent and created using organizational rules and regulations.
- 96. How can you measure project execution?
 - a. We can measure project execution using Activity Monitoring, Status Reports, and Milestone
 - b. Checklists.
- 97. Tell me about some project management tools.
 - a. There are many types of management tools used as per the need for a software project. Some of them are Pert Chart, Gantt Chart, Resource Histogram, Status Reports, etc.
- 98. What are software requirements?
 - a. Software requirements are a functional description of a proposed software system. It is assumed to be the description of the target system, its functionalities, and features.
- 99. What is feasibility study?
 - a. It is a measure to find out how practical and beneficial the software project development will prove to the organization. The software analyzer conducts a study to know the economic, technical and operational feasibility of the project.
 - i. Economic: It includes the cost of training, cost of additional and tools and overall estimation of costs and benefits of the project.

- ii. Technical: It evaluate technical aspect. Is it possible to develop this system?
Assessing the suitability of machine(s) and OS on which software will execute, knowledge of the software development and tools available for this project.
- iii. Operational: Here the analyst need to assess that the organization will able to adjust smoothly to the changes done as per the demand for the project. Is the problem worth solving at the estimated cost?

b. After, studying all this the final feasibility report is created.

100. What are functional and non-functional requirements?

- a. Functional requirements are functional features which are expected by users from the proposed software product.
- b. Non-functional requirements are related to security, performance, look, and feel of the user interface.

101. What is software metric?

- a. Software Metrics offers measures for various aspects of software process which are divided into:
 - i. Requirement metrics: Length requirements, completeness
 - ii. Product metrics: Number of coding Lines, Object-oriented metrics, design and test metrics.

102. What is modularization?

- a. Modularization is a technique which is used for dividing a software system into various discreet modules. That is expected to carry out the tasks independently.

103. What is cohesion?

- a. Cohesion is a measure that defines the intra-dependability among the elements of the module.

104. Mentions some software analysis & design tools?

- a. Some of the most important software analysis and designing tools are:
 - i. Data Flow Diagrams
 - ii. Structured Charts
 - iii. Structured English
 - iv. Data Dictionary
 - v. Hierarchical Input Process Output diagrams
 - vi. Entity Relationship Diagrams and Decision tables

105. What is mean by level-0 Data flow diagram?
- Highest abstraction level is called Level 0 of DFD. It is also called context level DFD. It portrays the entire information system as one diagram.
106. What is the major difference between structured English and Pseudo Code?
- Structured English is native English language. It is used to write the structure of a program module. It uses programming language keywords. On the other hand, Pseudo Code is more like to the programming language without syntax of any specific language.
107. What is structured design?
- Structured design is a conceptualization of problem. It also called solution design and which is based on 'divide and conquer' strategy.
108. What is functional programming?
- It is a programming method, which uses the concepts of a mathematical function. It provides means of computation as mathematical functions, which also produces results irrespective of program state.
109. What is Quality Assurance vs. Quality Control?
- Quality Assurance checks if proper process is followed while developing the software while Quality Control deals with maintaining the quality of software product.
110. What are CASE tools?
- CASE means Computer Aided Software Engineering. They are set of automated software application programs, which are used to support, enhance and strengthen the SDLC activities.
111. Which process model removes defects before software get into trouble?
- Clean room software engineering method removes defects before software gets into trouble.
112. Solve this problem
- There are twenty different socks of two types in a drawer in one dark room. What is the minimum number of socks you need to take to ensure you have a matching pair?"
 - If you pick up three socks, they may be of the same type even if the dds are 50%. Odds never an equal reality. Therefore, the only way to 'ensure you have a matching pair' is to pick up at least 11 number of shocks.
113. How you can make sure that your written code which can handle various kinds of error situation?

- a. I can write tests that define the expected error situations.
- 114. Explain the differences between a Thread and a Process?
 - a. A process is instance of the computer program. In a single program it is possible to have one or more threads.
- 115. Tell me the difference between an EXE and a DLL?
 - a. An exe is an executable program while a DLL is a file that can be loaded and executed by programs dynamically. It is an external code repository for programs. As both are different programs, reuse the same DLL instead of having that code in their file. It also reduces required storage space.
- 116. What is strong-typing and weak-typing? Which is preferred? Why?
 - a. Strong typing checks the types of variables at compile time. On the other hand, weak typing checks the types of the system at run-time. Among them, Strong typing is always preferred because it minimizes the bugs.
- 117. Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming.
 - a. Interface programming is contract based.
 - b. Object-oriented is a way to write granular objects which have a single purpose.
 - c. Aspect Oriented Programming is to segregate the code in such a manner that various objects carry the main tasks, and the subsidiary tasks are carried by independent objects.
- 118. Why using catch (exception) is always a bad idea?
 - a. It is a bad idea because:
 - b. As there is no variable defined, it is not possible to read the exception
 - c. It's good to use an exception when you have known exception types.
- 119. What type of data is passed via HTTP Headers?
 - a. Script and metadata passed via HTTP headers.
- 120. How do you prioritize requirements?
 - a. First, you need to design a system by evaluating data structure. Then you should move on to the code structure needed to support it.
- 121. Give me differences between object-oriented and component-based design?
 - a. Object-oriented design can easily be encapsulated to some degree in component-based design.

122. When do you use polymorphism?
- a. Polymorphism is used when there is a need for override functionality when inheriting class. It's about shared classes and shared contracts.
123. What is the difference between stack and queue?
- a. Queue is always First In, First Out
 - b. Stack is always Last In, First Out
124. What is essential for testing the quality of the code?
- a. According to me, the unit testing framework is essential for testing the quality of the code.
125. Do you think that the maintenance of software is expensive?
- a. According to me, maintenances of software will never be expensive if we are using proper development process.
126. Give me differences between tags and branches?
- a. Tags are for versioning releases which are temporary holding places for doing such thing. However, branches are deleted when those changes are merged into the trunk.
127. Where is a protected class-level variable available?
- a. Protected class-level variables are available to any sub-class derived from the base class.
128. Is it possible to execute multiple catch blocks for a single try statement?
- a. Yes. Multiple catch blocks can be executed for a single try statement.
129. When do you need to declare a class as abstract?
- a. We should declare a class as abstract in the following situations:
 - b. When the class is inherited from an abstract class, but not all the abstract methods have been overridden.
 - c. In the case when minimum one of the methods in the class is declared as an abstract.
130. Develop an algorithm that output your current location and a list of ATMs locations in that area. Get you the closest K ATMs to your location