DOWNLOAD PDF CONDUCTING THE JAVA JOB INTERVIEW

Chapter 1 : Need a trainer for conducting soft skills and interview | Call Center | Java | Training

*Conducting the Java Job Interview: IT Manager Guide for Java with Interview Questions (IT Job Interview series) [Jeffrey M. Hunter, Don Burleson] on www.nxgvision.com *FREE* shipping on qualifying offers. Offering accumulated observations of interviews with hundreds of job candidates, these books provide useful insights into which characteristics make a good IT professional.*

Most interviewers are unarmed with the appropriate interview questions , unable to follow-up up on candidate responses and otherwise fill the void that often results when two strangers sit together for the first time. In fact, many hiring managers fear interviewing and hiring. Still, the stakes are exceptionally high in terms of building your team and furthering your own success. On closer analysis, the list is fairly intuitive. Second, look for compatible communication and work styles. What makes you stand out among your peers? What would your most respected critic say of your strengths, areas for development, and future potential in your field? How many employees does your company have? How is your department structured in terms of reporting relationships? Look for Compatibility, not Just Likeability We all tend to hire in our own image, but look beyond immediate chemistry by asking questions like: How many hours a day do you find it necessary to work in order to get your job done? How sensitive are you to accepting constructive criticism? How much structure, direction, and feedback do you generally prefer on a day-to-day basis? Do you generally ask for permission or forgiveness when making decisions? A natural follow-up to this initial question might be: Tell me about a time when you may not have erred on the side of caution when you should have. Try questions like these to isolate those who are hungriest for the opportunity that you offer: Why would you want to work here, and what do you know about our organization? What makes us stand out in your mind from our competitors? If you were to accept this position with us today, how would you explain that to a prospective employer five years from now? In other words, how would this role with our company provide a link to your future career progression? After all, every relationship gives us an opportunity to share our wealth of knowledge and experience with others:

Chapter 2 : How to Conduct a Fair and Meaningful Technical Interview

Just like the title regarding the J2EE job interview, this book will help developers come up with quick answers and the proper vocabulary to sound proficient in a job interview, but it is not going to help you much in solving problems though software development questions.

How I ended up conducting the most successful technical interviews with a single question The hiring process A big part of my previous job was to take part in the hiring process and conduct technical interviews. That process was quite forward: Interns only went through 2 while the others went through all three steps. It was an enriching learning process for me, and I ended up figuring it out one step at a time. Now it is important for you to note that this occured in France, where you simply cannot fire people. Hire the wrong guy and you will be stuck with that person forever. It is critical to filter out the best candidates and not make any mistakes about it. It was a tedious process, and I loved every single part of it. The very-specific lottery quiz I conducted my first tech interview in At that time, the company already had a working process that I followed: The candidates would have 30 minutes to answer a 15 questions quiz. Then we would spend 15 minutes talking about their answers plus an additional 15 minutes answering questions about the job. I quickly realized how terrible that questionnaire was. It contained 5 trivial questions and then 10 very hard questions specific to the java frameworks we were mostly using: It went from - What is the difference between a class and an object? Pretty ironic for the interviewer Anyways I would usually skim really fast mins through their answers and spend the rest of the time talking about their resume. It sucked big time and I wanted to improve it. So I went online and compared hundreds of interview questions. At that time I believed in the quiz format. It just had to contain the right questions in order to reveal how good people were. The right quiz for the right people. The very generic quiz After about a month of research, I had come up with the best 50 questions I could find online. I felt that they were good questions because they could be answered in any language, and they were in a smooth crescendo of difficulty. I scattered the 50 and ended up with 5 sets of 10 great questions that I would hand out randomly. This was much better. The questions were good and I would also usually get good answers in return. Yes it did test whether or not the person knew programming theories, but in the end it left me clueless as to whether or not the person could code. The more I thought about it, the more I realized that there were 2 big issues with those 5 questionnaires: There was no way I could know if they were good programmers or not. What I really needed was a lot more questions, and questions specific to the job the person was applying to. Quiz manager This is where things got a little out of hand. I went ahead and created with the help of an intern a fully automated quiz tool: The tool made the hiring process perfect: It would then draw little graphs, generate and email reports to HR, displaying the results compared to the average along with a bunch of other useless metrics. Man was I proud of that tool! I was looking forward to having candidates take the test! QM made all of our lives so much easier, it seemed perfectâ€¦ Until we tested it on our own developers! Turns out that a lot of our great developers were getting the same score as some of the people I had refused. I had spent so much time building the tool that it took me a long time to realize the big mistake I had made: The user could only select one answer, and the questions ended up being mostly trick questions. The outcome was that we were not testing software development skills at all! It was tough for me to swallow my pride, but in the end I admitted that the tool was counter-productive, reflecting the wrong impressions about developers more than anything. I did some more research and checked at how some US companies did their screening process. This is when I decided to go for another method: Quite logical when you think about it Having learned some lessons with the first months, the test became quite simple: I would give out 3 algorithms to be solved in 30 minutes. Candidates could pick the language of their choice and have access to a machine disconnected from the internet. Those were classic problems found online: Everything got much clearer and better. I could directly see those who was indenting, commenting, using conventions, finding the solutions, etc. It gave me a pretty good sense of how much programming the person had done in the past. I like to think that candidates were comfortable with those tests because I had tried to take off all the pressure out of them. They could take their time, choose the language they wanted, ask for advice, etc. I was initially happy with the results and did

that for a couple of months. But were they really the great programmers I was looking for? The one question to rule them all I can remember exactly when I first started programming. It contained its own help screen with all of the functions and keywords of the language, like the perfect offline man page. To this day I distinctively remember the feeling that grasped me every time I hit F5 and saw my programs execute before my eyes. A single printed line, a prompt for a name, some colors, a puzzle I was in heaven. I remember putting line numbers before each command, filling my code with horrible GOTOs, learning with excitement and fascination something new everyday. I would spend hour after hour creating games, solving problems, showing stuff to my parents and friends. Well in fact I really sucked and my code was pretty horrible! But man did I love it!! I guess some people feel that type of adrenaline the first time they fly a plane, sail a boat, smoke weed, eat at in n out For me it was programming, compiling, executing. I gained that feeling 25 years ago, and it has never left me since. I was born for this. I have always been convinced that those who love code do not restrict their coding activities to their work. They take home that love and continue to create for fun as a hobby. How many times have I felt frustrated at work because of a struggling eclipse, only to find relief and joy when writing ruby on rails code back home! And so it was, that after 1 year of trial and error, I completely stopped handing out technical tests. I would sit down with the candidate, read and comment his resume without asking him any questions for a good minutes. And then I would flip over the resume, look at the candidate in the eyes and ask: Some answered vaguely about their previous work or school project. And then some others became suddenly alive and excited, even those who appeared to be the shyest. They would talk passionately about the game they were creating, the website they had made, the open source projects they had contributed to, the utilities they made after being stuck in the middle of nowhere without any internet access. They were proud to show me. I was always fascinated by what I heard and would ask about all the details of the project they had treasured. They opened up and talked about the technical difficulties that they had overcome, about the little personal touch they added. It was their baby. And as they talked it was impossible to miss: I could see that light in their eyes, the excitement of a child that compiles and runs his first hello world. I would know right then that we had something in common. They were programmers too. Yet once they got the job, they always ended up being golden developpers. They learned faster, they produced better code, they inspired others with their creativity and positivism. They were coders at heart. The question I would ask would include all types of projects, and I would put work projects at the exact same level as hobby projects. It was great but rare since they were changing jobs to meet people who were passionate about their work project. There is no perfect interview method, no method that will work perfectly for every single candidate out there. But I came to find out that even the best could freeze in front of a simple FizzBuzz problem, just because of the interview context pressure. So basically instead of asking the candidate if they knew "this" or "that", this question allowed candidates to bring me to their world, tell me who they were and what they knew best. I used those first minutes to chit-chat and make the candidate feel as comfortable as possible, by reading and making positive comments about the experience he had written down. My main objective was to spend most of the interview time listening to what he had to say on the thing he was most passionate about. I never looked down at candidates because each of them was unique and you never knew how good they were just by reading their resume.

Chapter 3 : The Best Interviewing Questions to Ask Job Candidates | www.nxgvision.com

Conducting the Java Job Interview has 6 ratings and 2 reviews. Howard said: The only good thing about this book is the question bank in the later half. A.

Having Trouble with the Technical Interview? Are you contemplating a job change? Are you ready to begin the interview process? Is this your first interview experience? Perhaps you have been through this process multiple times. Do you find the programming interview process intimidating and overwhelming? I have interviewed hundreds of Java developers and software engineers. There are numerous coding and non-coding questions that can be used to help indicate the quality of a particular software engineering candidate. Leveraging those experiences, I will outline a framework that will help you understand the ideal time to change jobs, provide guidance on which organizations to seek out or avoid, and then guide you through the preparation and interview process in a way that will help you best represent yourself when it is time to showcase your talents and skills. Preparation is the key to a successful coding interview. This book will help set the expectations on what things an interviewer looks for in a technical candidate. Interview Questions and Answers There are a number of questions that you should have answered prior to your next interview. You need to understand what motivations are driving your job search. You should know what kinds of questions an interviewer is likely to ask you, and what level of importance is applied to your answers to various questions and question types. While a Java developer would expect to see core Java questions, and a. Net developer would expect to see core. Net questions, there are a host of other topic areas that are important to the interviewer. You will find the following included in this book. Questions you should ask yourself when thinking about a job switch. Questions to ask your interviewer to help determine the organizational health of your potential employer. Characteristics of a great software engineer. Essential software engineer skills and competencies, both coding and non-coding related. The types of interview questions you may encounter. Checklist to help you prepare for your next interview. Interview questions you may be asked, and what the interviewer is looking for in your answers. Questions you should ask your interviewer, and the answers you should be looking for. Find Your eBooks Hereâ€¦.

Chapter 4 : conducting the java job interview | Download eBook PDF/EPUB

Conducting the Java Job Interview: IT Manager Guide for Java with Interview Questions by Jeffrey M. Hunter Offering accumulated observations of interviews with hundreds of job candidates, these books provide useful insights into which characteristics make a good IT professional.

Now you have to interview candidates for a marketing assistant position. Nine Tips for Interviewers Remember that during the interview process, candidates are deciding whether they want to work for you just as much as you are trying to decide whether to hire them. You have only about an hour to make a good impression on the candidate. Ask behavioural questions, as in "tell me about a time when you Previous successes are a good indicator of future performance. This may seem obvious, but by preparing your interview questions and reviewing the resume , you are showing the candidate you have taken the time to ensure a productive interview. Outline the interview structure for the candidate. First, give a brief description of the company, and then outline the job duties. Finally, ask the applicant questions. After that, the candidate will have the opportunity to ask you questions. This sets up the parameters of the interview, keeps you both focused, and gives the candidate an idea of what to expect. Dooney suggests hiring managers should talk only about 30 percent of the time. Allow candidates time to describe their skills and qualifications during the interview. Extend professional courtesies, says Dooney. Offer candidates a glass of water, and ask if they had difficulty finding the place. Consider giving them a tour of the office. Give them an opportunity to speak with other team members or prospective coworkers, if appropriate. Just as you are looking for eye contact and appropriate dress, the candidate is looking for those unspoken signals from you. Be sure your tone of voice is appropriate and professional. Dress as you normally would, and pay attention to manners. You are a representative of your company and department, so make sure your actions reflect this. If you spend the interview chatting, you may make a hiring decision because you liked the candidate versus whether the person is truly qualified for the job, he explains. This is one more way of extending a professional courtesy and gives the interview process closure. Could you use some help preparing? Join Monster for free today.

Chapter 5 : Interview Tips for the Interviewer | www.nxgvision.com

Offering accumulated observations of interviews with hundreds of job candidates, these books provide useful insights into which characteristics make a good IT professional. These handy guides each have a complete set of job interview questions and provide a practical method for accurately assessing the technical abilities of job candidates.

Kim How to Conduct a Fair and Meaningful Technical Interview When I began searching for my first job as a web developer, I applied to, and received interviews with, several companies. Some of my interviews were with Fortune companies; some of my interviews were with fledgling start ups. Regardless of the size of a company, the interview process was very similar: If the questions being asked were fair and meaningful, then I, regardless if I passed or failed the technical interview, would leave with a favorable impression of a company. At worst, I would leave without a job but with some new and useful knowledge. If the questions I was being asked were out of the scope for a particular position or merely a trick, then a company risked alienating me and other applicants. This person asked me to describe the difference between prototypal inheritance and prototipal inheritance. After my interview, I talked to a few of the other applicants, and we all agreed â€" we would never work for that company. Meaningful is considered asking questions that reveal some level of understanding of a fundamental concept. When a question is fair and meaningful, both the interviewee and interviewer benefit. I believe that both of these objectives can be meet with these three concepts: A rubric for grading a technical interview is located after these three concepts. Callbacks Interviewers should always ask an interviewee to define a concept. If the interviewer fails to ask this question, then the interviewee should volunteer to share their understanding of the concept. Without a mutual definition, the interviewee is unlikely to solve a given task. After a mutual definition is reached, the interviewer should present a question involving code: I want to explore your understanding of callbacks, so please create an implementation of a well-known function called reduce. At this point, the interviewer should present an invocation of reduce with example input and output data. This step enables an interviewer to understand how an interviewee thinks and to also prevent an interviewee from going too far down an incorrect path. The interviewee will create, based on my experience, an implementation of reduce using a for loop: Prompt the interviewee to refactor their implementation of reduce to include another well known function called each. This request will require the interviewee to use two callbacks, each nested inside of reduce: Ask an interviewee to define the concept of binding, ask the interviewee to create an implementation of bind, and ask the interviewee to talk aloud. In regards to bind, the interviewee can create an implementation with or without using a prototype. Interviewers should allow the interviewee to create the simpler implementation first â€" without a prototype. This approach enables the interviewee to build confidence when asked for the more advanced implementation. Event Emitters and Inheritance The concept of event emitters will be less familiar to an interviewee than callbacks and binding. Due to this reason, interviewers should clarify to the interviewee that many phrases are used to describe this concept, such as eventing system and eventing library. Once the interviewee has agreed to a mutual definition, present some restrictions for a desired implementation. An interviewer can achieve this goal with a prepared example of input and output data: I consider the following factors when I interview: Is the use of indentation or white space consistent? Are the names for variables descriptive? Is more than one use case considered? Has the interviewee defined the scope of a question? Is the applicant using native methods and not recreating them? Conclusion A technical interview can leave a lasting impression on an interviewee. If an interviewer can achieve this goal, the worst outcome for an interviewee is that they are not offered a job but they leave with some new and useful knowledge. Meet the author Cho is a full-stack web-application developer.

Chapter 6 : How to conduct a technical interview: 5 questions to ask - TechRepublic

Conducting The Java Job Interview: IT Manager Guide For Java These Senior Java Developer interview questions bring together a snapshot Similar job titles include Java Programmer, J2EE Developer, Java Can they show a wider.

Chapter 7 : For Managers Only ? Conducting the Programmer Job Interview

Conducting the Java Job Interview: It Manager Guide for Java with Interview Questions has 1 available editions to buy at Alibris. 10% Off through Friday.