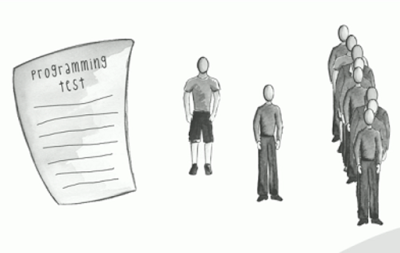
[**http://javarevisited.blogspot.com.by/2015/08/how-to-become-better-interviewer-programming.html**](http://javarevisited.blogspot.com.by/2015/08/how-to-become-better-interviewer-programming.html)

11 Tips to become a better Interviewer? (Programming)

For a change this article is not for candidates who are preparing for interviews, but for those programmers who are responsible for taking interviews. At some point in your career as programmer, you will have responsibility to hire developers for your company. This is a big responsibility and your ability to select right candidate will directly affect your company's future. This is more visible in small companies than larger organization where a pool of interviewer exists. In start-up if you are a technical lead and responsible for hiring, it's tough job. You have to hire right candidate which fits in companies culture and competent enough to do the job they were hired for. To be frank, conducting interviews is as hard as giving interviews. It's very difficult to know about candidate in 60 odd minutes and I am sure we all have missed out on some great programmers, who doesn't do very well on typical programming interviews. At the same time we end up hiring someone who looks brilliant on interview but fail to deliver. In this post, I am going to share some of my experience with you guys about taking interviews, which may help you to become a better technical interviewer and take effective interview.

## 11 tips to become a better Technical Interviewer

One of the truth about programming interview is that candidate prepare hard before giving interview, but how many of Interviewer prepare before taking interviews? We all rely on what we know, despite knowing it might not be latest and very quick to weed out candidates, who can't answer [typical weed out programming questions](http://javarevisited.blogspot.sg/2014/09/10-questions-to-make-programming-interviews-cheaper.html). In this article, I will share 11 tips to become a better Interviewer, as I said conducting interviewing is big responsibility, you become responsible for someone's career as well as your company's future, so take it very seriously. Hopefully this will lessen the popular sentiment that most programming interviews ARE a waste of time.  
  
  
1) First thing first, we **conduct interview to hire people, not to reject them**. Don't do interview if you are feeling angry or upset with something, with these kind of mindset you can't do justice with candidate. Instead of looking for positive things, you will look for negative aspect. You will probably look reasons to weed him out, consistently ask questions on his weak area to back your decision and cover yourself. Never do that, make every effort to give candidate proper chance to demonstrate their skill, ease them out, don't make candidates nervous.  
  
  
  
2) **Never insult candidates**, even if he is not able to answer a single question. I do couple of tech interviews every month and I regularly see people with very impressive resumes that [can't code Fizzbuzz](http://javarevisited.blogspot.sg/2015/04/fizzbuzz-solution-in-java-8.html). Instead, just give them X minutes (you don't have to tell them X) and then after X minutes simply say, *"OK... that's fine. You don't need to solve it here, but I just wanted to get an idea of your coding process"*, and either move on to next question or wrap the interview saying, do you have any questions for me?  
  
  
  
3) Always **give candidates opportunity to ask questions**, as much you are investing your time on conducting interviews, candidate is also investing his time. As much you are evaluating candidates, he is also evaluating you as potential colleague. You are not just hiring but also selling your Company to your next Rockstar or Superstar programmer. This is especially true with good candidates. They usually have more than offer at hand, and if you are not able to create repo for yourself and your company, there is good chance that he will consider taking your offer than your competitors.  
  
  
  
4) In order to bring best out of candidates, you can also **help him**. Whenever a candidate can't get on their own, I give progressively better hints until they come close. This helps you to gauge how many hints they need in comparison to other candidates and it also moves the process forward. Then move onto a different topic or skill. Remember, No one is good at everything or experienced in all fields. Many candidates will do poorly on one question but then nail the next one.  
  
  
  
5) **Speak slowly**. Yes it makes a lot of difference. When you are speaking with someone first time on a telephone interview, its difficult for both candidate and interview to understand each other, this is especially true if you are hiring for multi-national which has global hiring process. Many times candidate's accent is not clear or he might not able to understand your accent. I have often seen people calling one thing differently e.g.  char or char, SQL or Sequal. Speaking slowly helps both of you to understand each other better.  
  
  
  
6) **Always ask surprise questions**, put them in some pressure. You may be thinking that this is contracting with my earlier point, where I have mentioned that you should help candidate to feel better, but, remember you also have to save your company's interest. You must test candidate's problem solving skill and how he react in pressure situation e.g. when he knows nothing about how to solve problem but he has to solve it. These situations are not hypothetical, in fact we all face it on daily basis.  
  
The idea behind an interview is to understand how proficient a candidate is in articulating their thoughts, and how well they can handle difficult problems. No good interviewer asks only simple questions. The hard questions are there for the interviewer to understand how experienced candidate is in handling unknown situations, and how he/she performs in a "virtual" stressful situation. Its not a litmus test, but probably the best an interviewer you can do in 60 mins. These skills only come by practice, and you quickly identify whether candidate has gone through them or not.

[](http://4.bp.blogspot.com/-cq7TSW19iJE/VdNCcyWwIGI/AAAAAAAADmc/IE-d0TwYuy8/s1600/Programming%2BInterview%2BTips%2Bfor%2BInterviewert.png)

7) **Must test fundamentals like data structures, algorithms, and programming logic**. Throwing out some puzzles is also a good idea. You may get hated for asking *how to remove a node from doubly linked list?* for a web developer position. I know they might not need this detail but I will rate high to a developer who knows fundamental better but light on API rather than a programmer who are good at API but doesn't know [how a hash table works?](http://java67.blogspot.sg/2013/06/how-get-method-of-hashmap-or-hashtable-works-internally.html)  
  
  
  
8) **Don't just focus on tech skills**, also take notice of their soft skill. If a candidate is an expert in certain field but not able to clearly explain his thoughts or ideas then it's not a good hire. Good communication skill is very important, don't expect them to be Steve Jobs or Bill Gates but they should be able to articulate their thoughts.  
  
  
  
9) **Look for candidates who pay attention to details and ask right questions**. Many programmers make mistake of directly jumping into solution without understanding problem first. They don't even focus on context and big picture. In real world, you never get 100% requirement, it's your experience and skill which is required to fill the gap. Just think about what happens when you go to a doctor and says you have headache? Before giving you any medicine, doctor does his due diligence in checking temperature, checking your eyes, mouth and asking some question, that is called diagnosis and it's very important part of solution.  
  
This is true in every profession, when you go a carpenter and order for a bed, he will ask about its size, height, design and other key details. Similarly if someone gives you a problem to solve, you must understand it, don't afraid to ask question. Even business guys expect technology to ask right questions. Most of the time, they don't know what they want and these discussion brings out real details. So look for candidates who can think through, can find out gaps in requirement and asks right questions. This is the one skill which separates an [average programmer](http://javarevisited.blogspot.com/2015/05/how-to-differentiate-between-average.html) to a good programmer.  
  
  
  
10) **Start with easy questions and move on to increasingly difficult ones**. Ask questions which includes problem solving with technical details and ask candidate to solve them loudly. Remember, process is just as important as the answer and it gives the interviewer a detailed view into candidate's thinking process and level of competency with the technologies he needs to use daily.  
  
  
  
11) **Never ask questions whose answers you are not sure** because if you don't know the answer then how would evaluate an alternate wording of the answer, or something which is very close but not quite the answer? In fact, you should be able to answer the question if candidate ask you (though this is personal choice, some people prefer not to answer and ask candidates to find it themselves after interview). I personally prefer to answer question if right candidate is asking it, because I believe interview is a two way process, it's not just you, who are evaluating candidate, but candidate also evaluates you as potential colleague or to gauge company's. Another reason, I like to answer the question I ask because I appreciate curiosity and learning desire of candidate. It also gives me an opportunity to find out if candidate is a good listener or not and whether he pays attention to details or not.  
  
That's all guys. I am not perfect on taking interviews and I also make mistakes, but I like to learn from my mistakes as well as my fellow programmer's mistakes. Remember, my favorite quote, "If you learn from your own mistakes, there is very little you can learn but when you learn from other's mistakes, there is no limit", So don't forget to share your thoughts. Does this tips even make sense or these are just too dumb?  
  
If you want to improve your technical knowledge for interview purpose, you can also benefit from these books, which many candidates read for preparing interviews :

* Programming Interviews Exposed: Secrets to Landing Your Next Job ([see here](http://www.amazon.com/dp/1118261364/?tag=javamysqlanta-20))
* Algorithms For Interviews By Adnan Aziz and Amit Prakash ([see here](http://www.amazon.com/dp/1453792996/?tag=javamysqlanta-20))
* Top 10 coding interview problems asked in Google with solutions ([see here](http://www.amazon.com/dp/1482799014/?tag=javamysqlanta-20))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*