

Design Pattern Interview Questions

Looking into Interviews a little bit

The right attitude, level of confidence, knowledge, experience, soft skills, personality for the right job.

How candidates are screened and selected for interviews?

1. In-house recruiting team – Human Resources. They are non-technical people. They use **resume relevance** as the yard stick to measure candidates' suitability against a job. They work under stress to fill open positions.
2. From recruiters. Some of them have technical background. They understand a resume better and would do a preliminary interview before sending a candidate to the employer.

First impression:

1. How the person looks and behaves? (**Professional, confident, honest**, shy, nervous/timid, too relaxed, arrogant, strange)

Interview process:

3. To find out how smart, **knowledgeable, experienced**, 'talkative', relevant, hands-on, etc. the person is.
4. Some questions are used to test how knowledgeable/experienced/hands-on someone is while others are like "little windows" through which to see more about the person – **personality**, habit (for learning or how deep one delves into any issues), **potential**, etc.
5. For some other questions, they assume you have the knowledge but want to see how good you are in explaining or presenting something (or yourself).
6. If the person does not have the exact experience, is he/she a fast learner or interested to learn. (there is a big learning component for every new joiner for any job)
7. What else the person can do later.
8. Depending on the job, for example a very technical position, one approach is 'keep digging until the guy fails'. Your failing point tells more about the depth of your knowledge, rather than you failed answering the question.

Final impression:

9. **Good fit** for the job. Want to offer right away in fear of losing the candidate.

10. Average. Ok but if get a better candidate, will not consider this one.
11. Mismatch (salary, location, role, missing skill/experience, weakness in key areas, etc).
12. Comments on candidate to be provided if through a recruiter.

Practice questions:

1. General ones
 - a. What is a design pattern? Why design patterns are important?
 - b. What is an anti-pattern? Give an example.
 - c. Types of design patterns according to GoF classification?
 - d. What design problems do they try to solve for each type?
 - e. 3 examples for each type.
 - f. What patterns are useful for building distributed systems?
 - g. What is the best way to generate an application-wide unique object id? For example, in a software system that manages university applications/admissions?
2. Singleton
 - a. Explain how you implement a Singleton.
 - b. What is Lazy instantiation in an implementation of the Singleton Pattern?
 - c. How to make your implementation thread-safe?
 - d. If you used a synchronized method, why is it not ideal?
 - e. Or why is a synchronized method expensive? How expensive can it be compared with an unsynchronized method?
 - f. If you were asked to implement the synchronization mechanism in Java, how would you do it to keep the performance penalty minimum?
 - g. If can make it here, you are good but still they can dig even deeper...
3. Adapter
 - a. When to use the adapter pattern?
 - b. Have you used it in one of your projects? Describe it with more detail.
4. Proxy
 - a. What is the role of the stub-skeleton that are generated by the rmic compiler?
5. Template Method
 - a. Explain what it is.
 - b. What is the advantage of using it?
 - c. Why is it (the advantage) important to us?
6. Command
 - a. Why we store a request/action in a command object?
 - b. Give an example of using the Command pattern in a real life scenario?
7. Factory Method and Abstract Factory

- a. Explain the Factory Method.
- b. What is the advantage of using it?
- c. Explain Abstract Factory.
- d. What is the difference between the 2?

8. Strategy and State

- a. Can you compare the 2 patterns?
- b. Shall we use an abstract class or an Interface in Strategy/State Pattern?

9. Observer

- a. What is an Observer design pattern?
- b. How does it work?
- c. When do you want to use it in a real life project?

10. Others

- a. What key pattern/patterns would you use to implement an event driven architecture? Why?
- b. What key pattern/patterns would you use to implement a broker architecture? Why?
- c. What key pattern/patterns would you use to implement a pipe and filter architecture? Why?
- d. What key pattern/patterns would you use to implement a MVC architecture? Why?
- e. What key pattern/patterns would you use to implement undo/redo? And explain how?
- f. The 23 GoF design patterns help us solve a lot of problems. Can you think of any scenario in software design where they are less helpful? (that gives us an idea of the strengths and weaknesses of GoF patterns in software development)
- g. Can you name a few design patterns used in standard JDK library?