Valid if interviewing for a backend engineer or full stack software engineer

Note that this document just covers the code design and not the architectural design of the system.

Pre-requisites :

Object-oriented design is the process of planning a system of interacting objects for the purpose of solving a software problem. It is one approach to software design. The reader of this document is expected to know the difference between classes and objects and understand basic pillars of OO design ie. Encapsulation, Abstraction, Polymorphism and Inheritance.

First Step :

1. **Extremely important ::** Make sure you have brushed up on basic design patterns along with examples given in the article :
    1. Singleton (http://www.oodesign.com/singleton-pattern.html)
    2. Factory Pattern (http://www.oodesign.com/factory-pattern.html)
    3. Builder Pattern (http://www.oodesign.com/builder-pattern.html)
    4. Pub Sub (http://www.toptal.com/ruby-on-rails/the-publish-subscribe-pattern-on-rails - the article is rails specific but covers the essential idea)
2. **Bonus ::** Read up basic of everything on the front page here: http://www.oodesign.com/
3. Cover design principles, make sure you cover **SOLID** design guideline. (**S**ingle responsibility, **O**pen-closed, **L**iskov substitution, **I**nterface segregation and **D**ependency inversion) - http://www.oodesign.com/design-principles.html
    a. Please click into each link and try to see examples to get better understanding.

Basic checklist

1. Ask a lot of questions about requirements. Make sure you have defined scope of the task well before you start designing. OO design problems are often very open ended.
2. Make sure components in your design are modular
3. That you are using inheritance to avoid duplication.
4. Using composition where applicable.
5. Using pub sub where applicable.

DO :

1. Be receptive to hints from the interviewer. If you are on the wrong track, a lot of times the interviewer would give you hints / suggestion. Do not get defensive. Instead try to evaluate if the suggestion makes the design cleaner.
2. Drive the discussion. The more senior the role, the more important it is to drive the conversation.
3. Consistently have 4 things :
    a. The name of the key components, or things, in the system.
    b. The purpose, basic function, or main responsibility of each thing.
    c. A drawing of the key communication paths between them.
    d. A list of the services provided.

DONT :

1. **Data redundancy :** Make sure you are not storing the same data in multiple objects. Use inheritance / abstraction to avoid it.

Questions(Java oriented) [Recommended if you are a java eng] :

1. Give an example where you prefer abstract class over interface ?
2. When do you overload a method in Java and when do you override it ?
3. Why is access to non-static variables not allowed from static methods in Java.
4. What is Singleton design pattern in Java ? write code for thread-safe singleton in Java
5. check more here: http://javarevisited.blogspot.in/2012/06/20-design-pattern-and-software-design.html

Case Study 1:

Design an OO parking lot. What classes and functions will it have. It should say, full, empty and also be able to find spot for Valet parking. The lot has 3 different types of parking: regular, handicapped and compact.

Solution: Check
http://stackoverflow.com/questions/764933/amazon-interview-question-design-an-oo-parking-lot
The first answer covers all aspects of design well.
Note: Very common OO Design problem asked in interviews.

Case Study 2:

Design for a game of chess.

follow up question: How would you add support to records the progression of the game, undo a move etc.

Solution: https://inst.eecs.berkeley.edu/~cs162/sp07/Nachos/chess.shtml gives guidelines on how to start with the solution. Build the complete solution on your own.

Case Study 3:

Design a coin flipping game.

https://www.youtube.com/watch?v=fJW65Wo7IHI

Case Study 4:

Design a basic set of objects/methods to be used to simulate an elevator bank (scheduling of multiple elevators). What are the objects and their attributes/methods?

http://stackoverflow.com/questions/493276/modelling-an-elevator-using-object-oriented-analysis-and-design

Case Study 5:

Design a Caching Library.

1. Define interface for the library.
2. It should be possible to use any backend data storage (MySQL, MongoDB)
3. Should have configurable cache size.
4. What eviction policy will you use? (Configurable ?)
5. How will you design the library in a way that is can take a custom eviction policy from user of the library. (advanced)
6. Support eviction notification(advanced). *hint: use pub sub.*

Case Study 6:

Build Tic-Tac-Toe as a multiplayer online game using object oriented principles.

1. The game can be played for n x n squares.
2. The games rules should be separate from the rest of the application.

This problem has a lot of scope and can cover a lot of technologies. eg.

1. choice of server stack for a large scale online game
2. WebSockets
3. Long Polling
4. Pub Sub
5. IO driven server execution.

Case Study 7:

Design a coffee machine which makes different beverages based on set ingredients. The initialization of the recipes for each drink should be hard-coded, although it should be relatively easy to add new drinks. The machine should display the ingredient stock (+cost) and menu upon startup, and after every piece of valid user input.
Drink cost is determined by the combination of ingredients. For example, Coffee is 3 units of coffee (75 cents per), 1 unit of sugar (25 cents per), 1 unit of cream (25 cents per).
Ingredients and Menu items should be printed in alphabetical order. If the drink is out of stock, it should print accordingly. If the drink is in stock, it should print "Dispensing: ". To select a drink, the user should input a relevant number. If they submit "r" or "R" the ingredients should restock, and "q" or "Q" should quit. Blank lines should be ignored, and invalid input should print an invalid input message.