**Section 1 - (5 mins)**

1. Explain the difference between an abstract class and an interface.  
   **Answer:** Abstract class has some implementations and also it can have some variables in it but in Interface, we are only allowed to define the signature of methods. There is no method body definition and we only define inputs and outputs. Another difference is that an Abstract class can implement an Interface but Interface can’t implement an Abstract class.
2. What is the purpose of getters and setters in a class?  
   **Answer:** The purpose of having getters and setters is to monitor and change the behaviour of a variable on the outside world. A behavior of variables internally can be different from what is meaningful for other classes. We can change it by getters. Also for setters, we can translate meaning from another class to our internal meaning.

**Section 2 - Testing - (5 mins)**

1. Explain the purpose of black box testing

**Answer:** The purpose of black-box testing is to make sure everything is working as expected by knowing the inputs and outputs, we don’t have any clue about the functions and structures inside. We only know what are the inputs and what are the expected outputs. So we test it in order to test the overall functionality.

1. In your opinion what are the benefits of test cases

**Answer:** I assume you meant ‘unit test’. The benefits are as follows:

1. We will be sure that we have tested the functionality of the functions and classes.
2. We will be sure that most of our code is being executed and nothing’s wrong(I assume we have written enough test cases)
3. If we don’t write test cases for each class it’s really hard to test all functionalities in the runtime. It’s much better to test it unit by unit. But it’s even better and easier to write them down by writing unit tests.
4. Unit tests are reusable codes. So if anybody new joins a team can easily read the source code of unit tests and understands the functionality.
5. If a developer changes a code that has a bad impact on other dependencies or functionalities the tests will be failed. And it’s very important that the developer will understand and fix the error in the development time, not on production on massive stress.

**Section 3 - NodeJS specific questions - (5 mins)**

1. What is an error first callback, and what is the reason for this pattern existing?

**Answer:** an error first callback is a pattern that is used in most of nodejs async functions. When you call a function it accepts a callback function that the first parameter of it is an error that might happen during execution and the second parameter is the data as result.

I think(not sure) that the reason behind this pattern is that to remind developers that handling errors are always important: If you’re writing a callback function always check the error first.

I think with this pattern we don’t worry whether it’s the second parameter or it’s the first. And we will never be confused everybody uses the same style.

1. Explain the difference between fs.readSync and fs.read (File System module in Nodejs)

**Answer:** fs.readSync blocks the event loop in Nodejs so no asynchronous code can be executed. But with fs.read event loop will not be blocked and you can run other asynchronous codes on event loop too.

1. What tasks do you feel should be done asynchronously?

**Answer:** I think tasks that are using other resources such as network, io should be executed asynchronously. Even tasks that have network dependencies in it. For example, a MongoDB task wrapper should be defined asynchronously as it’s using network in it.