**1) What accomplishment are you most proud of and why?**

My career in the US Navy! I served my country as well as people of every nation around the world. I provided for my family and my wife was able to stay home and raise our kids, and I worked with an amazing society of people who are all as close as family yet as diverse as the entire globe. I worked my way up from new recruit all the way to being a Chief Petty Officer.

**2) Why do you want to work for Unity?**

I want to be part of an organization that is helping people around the globe to create what they've envisioned. I've heard that Unity will also help me grow my programming skills and let me move into more of a developer role as my skills improve.

**3) What are the boundary conditions of log(x)?**

x>0, x= Infinity

**4) You are given a function, f(x), which is storing a number, N, between 1 and 1024. When you call the function it will return:**

**a. -1 if x<N**

**b. 1 if x>N**

**c. 0 if x == N**

**What is the maximum number of guesses you would need to determine the value of N?**

9

**5) How many bugs can you release a product with?**

Ideally 0. Potentially Infinite (discovered + undiscovered). Realistically it is dependent on the severity of the bugs balanced against the shipping timeline, risk of taking a fix, work required to fix the bug, available dev resources, criticality of the product, and benefits of the product as it exists even with the bugs.

**6) What is the difference between an interface and a delegate?**

The signature of an interface must exactly match the implementation, but delegates may be implemented with different signatures so long as the inputs are valid.

**What do they have in common?**

They both only have a declaration and the implementation is created in a different object.

**When would you use one over the other?**

A delegate can be used for neater shorter code and where multiple implementations may be needed over the rigidity of an interface.

**Assignment 1: Car building**

**Q: Do you spot any obvious problems?**

While the write up specifies one or more wheels, the diagram appears to allow 0 or more wheels.

The write up specifies Exactly one carburetor but the diagram appears to allow 0 or 1 Carburetor.

Engines get carburetors, filling your car with a fuel/air mist seems like a bad idea. I'd have to recommend the carburetor be linked to the engine instead of directly to the car.

**Q: How does the design react to change?**

It allows you the choice of different engines, carburetors, and wheel types/numbers.

It doesn't seem to allow for electric motors, multiple carburetors, Fuel injection (no carburetors if you follow the write up vs. the diagram) or other possible future advances.

**Can we unit test it?**

Yes, the individual parts can and should all be unit tested as well as the car.

Unit testing against the car would show the inconsistencies between the design documents.