**Red Hat Open Ended Questions:**

**Q1**

I enjoy the challenge of learning new concepts and technologies. If I was to approach this situation, I would begin with the languages I need to use the technology, starting with the syntax and eventually moving to design patterns and libraries within the language. For example, I was working for a company doing word press sites and I had never used word press before so I just spent a day or two of my own time researching as much as I could about it as well as asking some of my co-workers when I wasn’t sure of a particular part of it. It is also very important to be patient in these situations and to understand that no one is going to know everything all the time. It is easy to get swayed by a lack of knowledge, but I try my upmost all the time to be the best I can be no matter the situation. While the theory is very important, I personally always learn more by doing something so I’m always happy to be thrown in the deep end as that’s where I perform best.

**Q2**

To me, a successful internship would be one where I have learned a great deal as a result. Personally, I am an avid student of technology and I always want to better my skills as well as my knowledge. Being able to contribute to actual projects is a success also as I believe theory can only bring an individual so far, often an individual needs hands on experience with a software/project. I would also like to be able to give something to the young students of today and hopefully provide them with some inspiration to pursue a career in Computer Science or a related field.

**Q3**

I was recently working on a project whereby I had to develop a guitar tuner web application. Some way through the project, I encountered a bug that would only recognize one of the strings being played and try to tune all strings to that note. As a result, the program would not work. At first, I had to take a break from it for 20 minutes as I needed fresh eyes going back into it. I started by console logging all the relevant information about the variables, it was there I discovered the algorithm I was using was not outputting the relevant indexes for my string array. I had identified where the problem was, now I had to figure out what part of the algorithm was faulty. Similar to before, I logged all variables and played each string separately until I found that the program was actually recognizing all strings but the mathematics I used to calculate the required string was wrong. At that stage, it was just a matter of altering the mathematics to increment up rather than down on the index. It was a simple typing error on my behalf but ended up breaking the program. Afterward, I ran some Boundary Value Analysis just to be sure and all was working well.

**Q4**

I was as a bartender and it the first time I was left in charge of the bar for the weekend. The bar was busy on the Friday night and I had assumed there was enough stock to do for Saturday and Sunday. The next night (Saturday), there was a party scheduled to start at 9:00PM. At 10:20PM, we had ran out of two of the most popular drinks. As a result, a lot of the people left and those who stayed had to change their drinks which was unfair to them. In hindsight, something as simple as double checking the kegs to make sure there was enough would have avoided this problem. The learning experience I gained from this was to never assume anything and always double/triple check everything before committing to an action.

**Q5**

Last year, I developed a software system for the sport sector. I was tasked to elicit a set of user stories, UML diagrams and develop a UI mockup. My idea was simple, a food diary that could tell the user specific information about how much they were eating and what was in it (i.e. number of carbohydrates, fat and protein). For years I had trained in powerlifting and I always wanted a user-friendly app that could help beginners as well as professionals so to me it was a no brainer to make a food diary app. I started with the user stories to get a good outline of the features I wanted to have. I eventually had a few more features such as a custom workout planner, online free workout plans and plans for certain competitions e.g. bodybuilding, powerlifting, 5k, etc. Next, I began to plan out the architecture of the app (based in java) in the form of UML diagrams using Lucid chart. This process was very insightful as to how software’s are designed on an architectural level. I then developed a mockup user interface using wireframe which would tie all the user stories and UML diagrams together. I particularly enjoyed this part the most. Lastly, I outline some basic system tests for the app such as Boundary Value Analysis and Branch Testing as well as some network tests to check the limit of the server.