Instructor Name:	Phillip Windley	
Course Name:	C S 462 Sec-001	
Response Rate:	25/40	63%

I am learning a great deal in this course.

Average Rating 4.9/8.0

	Response Count	**Response Rate(%)**
Very Strongly Agree:	0	0%
Strongly Agree:	5	20%
Agree:	7	28%
Somewhat Agree:	3	12%
Somewhat Disagree:	4	16%
Disagree:	3	12%
Strongly Disagree:	2	8%
Very Strongly Disagree:	1	4%

Course materials and learning activities are effective in helping me learn.

Average Rating 4.7/8.0

	Response Count	**Response Rate(%)**
Very Strongly Agree:	1	4%
Strongly Agree:	2	8%
Agree:	7	28%
Somewhat Agree:	6	24%
Somewhat Disagree:	1	4%
Disagree:	6	24%
Strongly Disagree:	0	0%
Very Strongly Disagree:	2	8%

This course is helping me develop intellectual skills (such as critical thinking, analytical reasoning, integration of knowledge).

Average Rating 5.2/8.0

	Response Count	**Response Rate(%)**
Very Strongly Agree:	0	0%
Strongly Agree:	6	24%
Agree:	7	28%
Somewhat Agree:	6	24%
Somewhat Disagree:	1	4%

Disagree:	3	12%
Strongly Disagree:	1	4%
Very Strongly Disagree:	1	4%

The instructor shows genuine interest in students and their learning.

Average Rating 6.6/8.0

	Response Count	Response Rate(%)
Very Strongly Agree:	4	16%
Strongly Agree:	8	32%
Agree:	11	44%
Somewhat Agree:	2	8%
Somewhat Disagree:	0	0%
Disagree:	0	0%
Strongly Disagree:	0	0%
Very Strongly Disagree:	0	0%

What is going well in class? What contributes most to your learning?

Labs are interesting and not all that difficult.

Projects are easy.

Dr. Windley is a very talented and smart computer scientist. I am really impressed with the infrastructure he has built for distributed systems programming. I assumed this class would be about the client/server architecture and the distributed nature of big companies (appserver, webserver, database server, etc), but I like the approach to separate networks working together in the internet of things. I also appreciate that Dr. Windley has the knowledge to answer lots of questions from us and that he takes the time to answer our questions.

Dr. Windley is passionate and engaging about the subject matter

Group activities, thinking about and designing conceptual systems in parts

I'm happy to be exposed to a new programming paradigm, i.e. an event-driven language KRL. I think it's the first time at BYU I've been required to use a language that isn't imperative/object-oriented. Thank you!

the most important things are that we can use any technology available to us. and that's great.

Dr. Windley is a good guy but I do not like this class. I expected to learn some very practical applications of Distributed Systems. The first couple of projects we worked on were great, exactly what I was expecting, but since then Dr. Windley has been having us learn his programming language KRL. I am not a fan of KRL and find no practical use for learning this language.

Learning how to use AWS was practical. Thinking in an event-driven paradigm is too.

The first two weeks were great. Dr. Windley shared some awesome stories, we talked about how these systems work and I found the material really interesting.

I learn most when we talk about current systems and the advantages/disadvantages of how they are built/designed

I quite enjoyed learning to set up an AWS instance. It is also interesting to hook up to and use various web services.

I like the labs. They are pretty simple (which is appreciated since we're only given a week to do them) but teach important concepts.

Explanations of real world experience and how this fits into the current view of distributed systems.

I like Dr. Windley's blog, he puts lots of interesting articles and meaningful material
Dr. Windley's experience and stories are very memrable.
I have been very impressed that we use AWS and have a chance to learn useful things that we aren't taught in other classes.

Dr. Windley is good at explaining about KRL - the language that we are learning about for the class.
Mostly the labs are what contribute most to my learning. That and the book. It's nice to come to class though and receive a little more about the assignment/reading material

Professor Windley is really good at teaching difficult concepts. I appreciate the experience I'm getting with new technologies, like AWS.

Discussion of not-so-traditional uses for the web which expand my understanding of the possibilities of what the web could become.

I don't feel like we've really learned much at all that is useful in the real world. All he has taught us is his how to use a language he invented on a platform he owns using a website he created.

The class isn't overbearing and the class does have important topics that are covered.

Walking through different examples. Doing the labs. Stories and personal experiences help me remember and internalize different concepts and ideas.

The labs.

The design exercises are helpful.

I feel like the professor is good at teaching us about future possibilities in the software industry

What could be improved? How could this course be more effective in helping you learn?

Documentation on KRL is really poor and hard to follow. Examples are not clear as to what goes where in the code. often they are one-liners that are given out of context.

I am having a hard time connecting the dots. I know that evented systems are important, but I don't feel that the projects with KRL are helping me understand distributed systems any more than I already know.

We have been talking a lot about the KRL language and although a knowledge of the language is helpful, I would like to learn about how distributed systems actually work with each other. How they communicate, how to push data from one to another, polling problems, etc.

I feel like it is quite good.

Although KRL is built with event systems in mind, playing with things like RabbitMQ, pubsubhubub, beanstalkd would be closer to what we'd use in a real career situation

The course isn't what I expected a distributed systems designs class to be. I was expecting something like what is described at MIT's Distributed Systems class website (http://css.csail.mit.edu/6.824/2014/): "[This course] will present abstractions and implementation techniques for engineering distributed systems. Major topics include fault tolerance, replication, and consistency."

Rather, I feel like I'm *primarily* learning a domain-specific language with very limited use (which I don't mind learningâ€"I just wish it wasn't the main focus) and how to call on third-party APIs. It seems I'm being taught how to use and rely on *other people's* distributed systems, but I thought the course would teach something about how distributed systems are designed and built.

if we actually learned about developing distributed systems rather than learning KRL.

We could stop using KRL and use whatever language we want (or at least a subset of professor defined languages) to implement the projects.

We learn Dr. Windley's thought child (the KRL programming language) instead of Distributed Systems. KRL skills are not useful outside of the class or Dr. Windley's company, Kynetx.

I think the focus is too much on KRL. I feel like almost all the lectures are just reviewing the API and learning the language. I would much rather talk about purposes for this type of system. More on Web Servers, Databases, Applications, that are distributed etc.

I feel like I am not learning anything about distributed system. All I am learning is KRL. I would much rather learn about systems, or, "Designing and implementing client-server enterprise applications. Web servers, application servers, database connectivity, remote procedure calls, transactions, messaging, directory, naming services, threads, security, data formats." Out of all of these concepts on the course description, we only learned about a small number of these things, and only during the first two weeks of the course.

I would like to devote more time to commonly used languages and systems rather than the rarely-used KRL and Kynetx.

Sometimes it has been difficult for me to focus during lectures. If what we're talking about in lectures doesn't relate to the lab I'm working on then I have a hard time paying attention. That's obviously my fault as much as anyone's, though.

Larger projects that build on each other.

We focus alot of time on learning to program in KRL. I feel like I don't yet understand the bigger picture.

I feel that the classes don't do anything to keep the students there. Compared to the first day of class - when there was not enough seats for everyone, now there are multiple seats open due to students not caring about attending class anymore.

I wish we would focus a little more on things like how to efficiently maintain a multi-server web application. While KRL is interesting, I feel like there are more applicable topics we could be focusing on.

I have a coworker that took this class years ago and told me that everything he needs to do in his job he learned from this class. I have been surprised to see that I won't be able to say the same thing. I feel like this course should be named "Writing Evented Systems with KRL". We are giving KRL and the possibilities of an evented system all the attention. KRL and evented systems are definitely interesting and valuable to look at. However, I feel if a potential employer sees "Distributed Systems" on my list of courses, they will expect something different from what I'll be able to deliver. Sure, I can figure stuff out on my own, but I would have liked to have my educational experience prepare me to deliver what employers will expect from me. Even if I do end up developing an evented system eventually for some company, I'll probably leverage their knowledge of javascript moreso than force them to learn KRL, so it's hard for me to dive into the details of KRL for almost the entire semester when I don't feel I'll be using it in the future very much. I'm not saying don't use KRL, I just think it's one cool thing under the umbrella of distributed systems. I would like to see more of the other cool things under that umbrella so that I can make an educated decision about where to get my hands dirty with the details.

It could teach us how to use real, modern distributed systems.

The principles of the class make sense. KRL has been a negative experience so far in terms of doing the homework. It is expected for the first lab to be difficult with learning a new language. But it has been frustrating to design the lab in a way that works and then learn that KRL syntax won't allow it.

Sometimes the things on the projector screen are a bit small so it's hard to follow along since it's hard to see exactly what is going on. If the font on the screen was a bit bigger it would be easier to engage in what is going on.

I'm disappointed that this class has turned into a KRL class. There is a whole lot more to building web apps. A more useful exercise would be to build an evented distributed system, not use one built by someone else.

I think we spend a lot of time just talking about KRL, things we could learn on our own. I would enjoy more duscussion about other ways that distributed systems work.

Spread out the exercises throughout the semester