

Lecture 1: January 27

Introduction:

Favorite econ class?

What class did they learn the most in?

Bus230 projects?

Why are you taking this class?

Go over syllabus

Exams (Dates)

Project Discuss Timeline and due dates

Homework

Lab on Thursdays

Why will this class be interesting useful?

Freakonomics

Hal White's paper is the most cited paper in economics since 1970.

All questions are ultimately empirical

Emailed readings (On Measurement)

Required Readings: Stevens, S. S. (1946). On the Theory of Scales of Measurement. *Science*, 103(2684), 677-680.

Webpage: hyper stat online

<http://www.davidmlane.com/hyperstat/>

or Zikmund Chapters 20-23

An Interview with Google's Chief Economist Hal Varian

<http://freakonomics.blogs.nytimes.com/2008/02/25/hal-varian-answers-your-questions/>

Q: Your job sounds extremely interesting. What jobs would you recommend to a young person with an interest, and maybe a bachelors degree, in economics?

A: If you are looking for a career where your services will be in high demand, you should find something where you provide a scarce, complementary service to something that is getting ubiquitous and cheap. So what's getting ubiquitous and cheap? Data. And what is complementary to data? Analysis. So my recommendation is to take lots of courses about how to manipulate and analyze data: databases, machine learning, econometrics, statistics, visualization, and so on.

<http://googleblog.blogspot.com/2008/07/our-googley-advice-to-students-major-in.html>

Our Googley advice to students: Major in learning

7/15/2008 05:48:00 PM

Management guru Peter Drucker noted that companies attracting the best knowledge workers will "secure the single biggest factor for competitive advantage." We and other forward-looking companies put a lot of effort into hiring such people. What are we looking for?

At the highest level, we are looking for non-routine problem-solving skills. We expect applicants to be able to solve routine problems as a matter of course. After all, that's what most education is concerned with. But the non-routine problems offer the opportunity to create competitive advantage, and solving those problems requires creative thought and tenacity.

Here's a real-life example, a challenge a team of our engineers once faced: designing a spell-checker for the Google search engine. The routine solution would be to run queries through a dictionary. The non-routine, creative solution is to use the query corrections and refinements that other users have made in the past to offer spelling suggestions for new queries. This approach enables us to correct all the words that aren't in the dictionary, helping many more users in the process.

How do we find these non-routine savants? There are many factors, of course, but we primarily look for ...

... **analytical reasoning**. Google is a data-driven, analytic company. When an issue arises or a decision needs to be made, we start with data. That means we can talk about what we know, instead of what we think we know.

... **communication skills**. Marshalling and understanding the available evidence isn't useful unless you can effectively communicate your conclusions.

... **a willingness to experiment**. Non-routine problems call for non-routine solutions and there is no formula for success. A well-designed experiment calls for a range of treatments, explicit control groups, and careful post-treatment analysis. Sometimes an experiment kills off a pet theory, so you need a willingness to accept the evidence even if you don't like it.

... **team players**. Virtually every project at Google is run by a small team. People need to work well together and perform up to the team's expectations.

... **passion and leadership**. This could be professional or in other life experiences: learning languages or saving forests, for example. The main thing, to paraphrase Mr. Drucker, is to be motivated by a sense of importance about what you do.

These characteristics are not just important in our business, but in every business, as well as in government, philanthropy, and academia. The challenge for the up-and-coming generation is how to acquire them. It's easy to educate for the routine, and hard to educate for the novel. Keep in mind that many required skills will change: developers today code in something called Python, but when I was in school C was all the rage. The need for reasoning, though, remains constant, so we believe in taking the most challenging courses in core disciplines: math, sciences, humanities.

And then keep on challenging yourself, because learning doesn't end with graduation. In fact, in the real world, while the answers to the odd-numbered problems are not in the back of the textbook, the tests are all open book, and your success is inexorably determined by the lessons you glean from the free market. Learning, it turns out, is a lifelong major.

Posted by Jonathan Rosenberg, Senior VP, Product Management