# CSC 484 Lecture Notes Week 1
# Introduction to the Course

I. Relevant reading.

A. Textbook Chapter 1.

B. Paper of the week (covering both weeks 1 and 2, due to short week 1):

*Investigating attractiveness in web user interfaces*

by Hartmann, Sutcliffe, and De Angeli, from the University of Manchester.

II. Go over first-day handouts:

A. Syllabus.

B. Questionnaire on areas of project interest and expertise.

C. Assignment 1.

III. Introduction to the class subject matter, and the text book (Preface, Chapter 1, Section 1.1).

A. To a large extent, the organization of the book will provide the framework for the lectures.

B. Following a recommendation in the book's preface, we will start with Chapters 1,9, and 12.

C. Subsequently, we will proceed in order of the remaining chapters.

IV. On good and poor design (Section 1.2).

A. "Good" means an interactive product has certain important traits, such as

1. easy to learn
2. effective to use
3. provides an enjoyable user experience.

B. There are some systematic ways to measure such traits.

1. We can rely on experts' judgment in the design and evaluation process.
2. We can conduct controlled experiments with users of a finished product.

C. The bottommost line is ***know the user*** of whatever interactive artifact you're developing -- before, during, and after the development.

V. High-level interaction design principles (Section 1.2.1).

A. Again, know your audience (cf. the bulleted items on Page 6 of the book).

1. The users *RULE*.
2. Know what they're good at and bad at.
3. Understand what they know and don't know.
4. Provide interface contexts they're familiar with.
5. Know how they currently do things.
6. Know what they like and dislike.
7. *Listen* to them and involve them *fully* in the interaction design process.
8. If in doubt, do things electronically the way they're are done non-electronically.
9. Always ask the user what's "aesthetically pleasing" and "elegant".

a. For example, the book authors did not know me very well as a user when they decided to cite the marble phone answering system as an example of good design.

b. I find the marble-based design childish and inelegant.

c. I would much prefer a digital screen with the instructions "You have 3 messages; press the ’Listen’ button to hear them."

B. The principle of least astonishment.

1. Simple tasks should be performable quickly, in a few number of steps.
2. More complicated tasks should be performable in some form, but it’s OK if they take longer.

C. Use "real-world" metaphors *judiciously*.

D. Treasure simplicity.

E. Be prepared to work with people who may have vastly different views than you do.

VI. Interaction design compared to software engineering (Section 1.3).

A. Everybody wants their own particular discipline to "run the show".

B. Software engineers may think that they’re role is central, or over-arching in the process of building a usable software product.

C. Interaction designers may well think the same thing.

D. In practice, having a *product manager* run the show can be quite effective.

1. The product manager carries the "vision thing" for what the product should be.
2. This person oversees and coordinates all of the people from the different disciplines shown in Figure 1.4.

VII. Interaction design (ID) and other disciplines (Sections 1.3.1 - 1.3.2).

A. As we’ll see next week, there is a good deal of similarity between the ID process and the SE process.

B. In both cases, product end-users should play a key role in the steps of the process that involve the design of the end-user interface.

C. The analogy to building architects and engineers is apt, from page 9 of the book:

1. Interaction designers are analogous to building architects -- they deal with people who’ll use the product.
2. Software engineers are an analogous to civil engineers -- they deal with proper product construction.

D. While today’s software engineers may see they’re job as both of these things, not everyone else does.

E. Overall, the book is good for computer scientists and software engineers, to help us broaden our perspectives.

1. CS and SE practitioners must keep in mind that software artifacts are deployed in many different settings these days, beyond desktop and web-based applications.
2. That said, the focus of 484 is necessarily more on the HCI aspects of ID, than on its other aspects.

a. Ideally, 484 could involve genuinely multi-disciplinary teams, but this is logistically infeasible given the current course structure.

b. There may well be some role playing in the 484 team projects, where selected team members will assume the duties of domain experts from areas outside of the computing disciplines.

VIII. The elusive "user experience" (Section 1.4).

A. It’s highly subjective and very personal.

B. There is yet no established science to measure, categorize, or analyze it.

C. In 484, you’ll get a chance to describe yours, and try to capture that of others.

D. You’ll start in Assignment 1.

IX. The process of interaction design -- ID (Section 1.5).

A. This is very much like the user-centered requirements acquisition process we followed in CSC 308.

B. When the book talks about "design", they’re referring to the interface design, not the underlying software model or program design.

C. When they refer to "building interactive versions of the design", they generally mean prototyping the behavior, rather than building an operational product.

D. The idea that the activities "inform one another and are repeated" is the same concept as iteration in a software development process.

E. A key difference between the ID process and the SE requirements process is that ID typically involves more explicit steps to analyze usability, throughout the process.

1. In the parlance of Fisher’s 308 process, usability analysis can be considered a *pervasive step* of the ID process.
2. Usability analysis is covered in Chapters 9-12, and will be a recurring subject of 484 assignments.

F. Another part of the ID process often missing from SE is the analysis of cognitive and social aspects of human behavior that affect the interactive experience.

1. This follows the "know your users" design principles.
2. Chapters 3-5 of the book focus on issues of understanding users.

X. ID goals (Section 1.6).

A. Usability goals -- how the product behaves

B. User experience goals -- how the user feels about it

C. Design principles -- how to achieve the goals

XI. Usability goals (Section 1.6.1).

A. *Effectiveness* -- how well does an interactive product do its thing?

B. *Efficiency measures* -- how long does it take to get something done?

1. The number of discrete interactions required to accomplish a task.
2. The amount of real time that elapses from initiation to completion of a task.

C. *Safety* -- how well does the product protect the user for doing (dangerously) erroneous things.

1. For software products, "dangerous" can include more mundane aspects of behavior than physical harm, such as loss of critical data.
2. In some cases, software can in fact be unsafe, in that its failures lead to physical property damage or harm to humans.

D. *Utility* -- the product does (just) what it’s supposed to do.

1. No missing critical features.
2. No clearly superfluous features.

E. *Learnability* -- how easy is the product to learn initially?

F. *Memorability* -- how easy is it to remember how to use, once learned?

XII. User experience goals.

A. These are highly subjective and personalized among users.

B. There is a laundry list of them on the top of Page 26 in the book.

C. The importance of such goals had been historically downplayed in HCI usability studies, until rather recently.

   1. This is likely do to the difficulty in quantifying such subjective usability characteristics, compared to the more specific usability goals like efficiency and utility.
   2. Even Donald Norman, a pioneer of cognitive science and user-centered design, has changed his way of thinking in this regard.
   3. New HCI research braves this frontier, such as this week’s 484 research reading on "attractiveness".

XIII. Design principles -- "dos and don’ts" of ID (Sections 1.6.3, and 15.2).

A. Those listed in Chapter 1 are quite intuitive:

   1. *Visibility* -- what you see is what you can do.
   2. *Feedback* -- let the user know what the heck is going on.
   3. *Constraints* -- limit interactions based on context.
   4. *Consistency* -- don’t do the same thing two different ways.
   5. *Affordance* -- let the user know how to use something by its structure or layout.

B. From Chapter 15, Nielson’s usability heuristics are a more specific version of the general design principles, and are the ones you’ll use in Assignment 1:

   1. Visibility of system status
   2. Match between system and the real world
   3. User control and freedom
   4. Consistency and standards
   5. Help users recognize, diagnose and recover from errors
   6. Error prevention
   7. Recognition rather than recall
   8. Flexibility and efficiency of use
   9. Aesthetic and minimalist design
   10. Help and documentation

C. There are lots of examples online, accompanied by lots of opinion.

   1. Nielson’s site is `useit.com`.
   2. Another site mentioned in the book is `asktog.com`
   3. There’s also `baddesigns.com`

D. Looking at lots of examples, and gaining your own experience by "doing" are how you’ll get good at ID.