Course Description

Explores the relationship between theories of design and computational tools. Explores how the emergence of computing as a mainstream tool in design is changing architectural practice. Discusses how, as with other technologies that have revolutionized the practice of architecture, information technologies carry hidden implications about design process and products.

This is a seminar about

- **Ideas** of computing and design and how they do or don’t interact, including good ideas that went nowhere while weak ideas prospered.
- **Design** – what it is, what it isn’t, what it could be, who does it, how they do it.
- **Computing** – is it a machine or a brain? Does it matter?
- The **Past** – influential concepts, theories and software in the domain of design computing.
- The **Present** – theories, trends, trajectories.
- The **Future** of design and its uneasy relationship to computing. Opportunities to explore.

Methodology

- **Reading** primary sources – papers, books, online – to understand the major themes in DC.
- **Discussion** with your peers about the ideas, sharing your interests and thinking.
- **Writing** your own thoughts, bolstered by reference to the readings, in a book report and research paper.

Goals for the quarter

- To **understand** the broad brushstrokes (and some details!) of thought at the intersection of computing and design, becoming acquainted with the important authors, ideas, and projects that have contributed the most to bringing us to the current time and place.
- To **take ownership** of the vocabulary of ideas and concepts necessary to participate in the larger worldwide discussion of design computing and to see what current topics are being talked about.
- To **focus** your thinking on a particular set of ideas and examine them in some detail, providing opportunity for you to produce a paper and lead a discussion with the other students.
Grading

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading &amp; discussion in class</td>
<td>40%</td>
</tr>
<tr>
<td>Book Report</td>
<td>20%</td>
</tr>
<tr>
<td>Research Paper</td>
<td>30%</td>
</tr>
<tr>
<td>Attendance &amp; Participation</td>
<td>10%</td>
</tr>
</tbody>
</table>

Expectations

- It is expected that you will actively participate in all activities related to the course.
- There will be regular readings assigned from an online collection, to be discussed in class as shown in the schedule. You are expected to read the material and come to class prepared to discuss. If you cannot attend due to illness, you are expected to post your ideas to the course BBS.
- You will also be expected to read a book outside of class by the middle of the term, to write a (minimum) three-page book report on it, and to make a brief in-class presentation about it. A list of books will be provided, from which you may choose (alternatives are possible, but must be approved in advance).
- Finally, you will be expected to prepare a 10-15 page research/position paper on the topic “Design Computing’s Grand Challenges” and present your paper at the end of the quarter.

In case of illness

This class meets just once a week. If you get sick and miss a day, that’s 10% of the class sessions (ouch!). Obviously the most important thing is to give yourself time to get better, but stay connected. Send me an email to let me know what’s going on. If you get to the “feeling better, but still not good enough to go to class” you can still do the readings and keep in touch via the class email list. Take the “Questions to think about” from the reading assignments and write a short essay addressing them and anything else you find interesting, then share it with us via the course BBS or mailing list.

Some words and people to check into

I realize that there is a very large body of material in the background of “design computing,” including famous projects, people, and terminology. It is unlikely that you are familiar with all of these. Each of us comes with a different background, and each of us can learn more. Take responsibility for learning something “extra” every day or two. Google and Wikipedia are great aids in this context, as are the PDFLIB and CUMINCAD. When you dig into something new, write up a short post about what you learned and send it to the mailing list (arch587a_au12@uw.edu). The extra-credit bean-counters will smile upon you.

Words:
- Pattern language.
- Optimization.
- Traveling Salesman Problem.
- Finite Element Modeling.
- State space.
- Visualization.
- Photorealistic imagery.
- Non-photorealistic rendering (NPR).
- Simulation.
- Human Computer Interaction (CHI, or HCI).
- Ubiquitous Computing (ubicomp).
- Parametric modeling.
- Building Information Modeling (BIM).
- Space syntax.
- Isovist.
- Fabrication.
- Embodied interaction.
- Problem Solving.
- Hill-climbing algorithm.
- Calm media.
- Passive awareness.
- Biomimetic.
- Emergence.
- Genetic algorithm.

People:
- Ivan Sutherland,
- Herb Simon,
- Christopher Alexander,
- William J. Mitchell,
- Charles Eastman,
- George Stiny,
- Don Greenberg,
- Hiroshi Ishii,
- John Gero,
- Mary Lou Maher,
- Chris Yessios,
- Ulrich Flemming,
- Michael Benedict,
- Malcolm McCullough,
- Donald Norman,
- Greg Ward,
- Paul Dourish,
- Tom Maver,
- Yehuda Kalay.