Programming as a Cultural Technique

Overview
Programming undergirds cultural, economic, and even juridical infrastructures: global finance, power grids, search engines, word processes, library databases, biometric databases, legal evaluation and decision-making, architectural drafting, urban planning, state administration, and cultural policy are informed by the analyses and relays of digital programs. Increasingly language, speech, images, and sounds—the music of Lady GaGa, the Ice Age franchise, and an avalanche of eBooks—are produced, processed, and analyzed by the software that is constantly selecting and discriminating among the possibilities and probabilities of expression.

Only in the 1980s did cultural theorists begin the serious work of reflecting on the rapport among programming and human culture. Early accounts by the likes of Ben Schniderman and Friedrich Kittler posited a strong opposition between the logical basis of programing and the phenomenological level of everyday human perception and interaction. In recent years a new trend has emerged in critical and aesthetic though on programming: Theorists of software studies (Manovich), critical code studies (Hayles), and Kulturtechniken (Pias), among others, posit that programming and culture interweave and rework an always already technical human culture.

Course Content
Departing from the perspective of recent theorists of Kulturtechnik (e.g. Kassung, Krajewski, Macho, Pias, Siegert), this course will offer an introduction to classic and recent cultural theories of programming. We will incorporate recent research in fields such as software studies and consider relevant political and cultural movements (e.g. the Pirate Party and the Free Culture movement). Students will complete the course with an understanding of the fundamental debates in contemporary cultural theories of programing as well as an ability to create their own elementary programs in Perl.

Each week we will meet twice: Once for a lecture and seminar discussion on aspects of the cultural history and theory of programming, and again in the computer laboratory for applied work in Perl programming. Lectures, seminars, and programming sessions will be complemented by use of an online forum for reflecting on and discussing the relationship between the theoretical and applied aspects of the course. For their final students may either write a program or a seminar paper.

Mode of Instruction
The course will be co-taught by Professor Dr. Christian Kassung and Dr. Bernard Dionysius Geoghegan. Readings and lectures will be in German and English, depending on the session. No prerequisite experience in programming necessary or expected.

Basic Texts
**Prehistory of Programming**
Excerpts from texts by Pascal, Leibniz, Jacquard, Babbage, Lovelace, Hollerith.

Supplementary readings:

**Recent Origins of Programming**

Supplementary readings:

**What is Programming? What is Programmable?**

Supplementary Readings:

**The Rise of the Interface**

**What is Software?**

**What is Software Studies?**
What is Hardware?

What is a Database?

What is Code?

The Politics of Code

Supplementary Readings:

Programing and Gender

Supplementary:

Can Programs Think?
424.

**Aesthetics of Code**
Hayles, Katherine N. “Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis.”
1, no. 1 (n.d.).

**Cultures of Coding**
Berry, David M. *Copy, Rip, Burn: The Politics of Copyleft and Open Source: The Politics of Open

**Cultures of Coding 2**
Kelty, Christopher M. *Two Bits: The Cultural Significance of Free Software*. Durham, NC: Duke
Pirate Party manifesto?

**Additional Readings**
Berry, David M. *The Philosophy of Software: Code and Mediation in the Digital Age*. New York:
Palgrave Macmillan, 2011.
Bogost, Ian. *How to Do Things with Videogames*. Electronic Mediations. Minneapolis: University of
Collins, Harry M. *Artificial Experts: Social Knowledge and Intelligent Machines*. Cambridge: The MIT
Dreyfus, Hubert L. “Response to Collins, Artificial Experts.” *Social Studies of Science* 22, no. 4
Row, 1979.


———. “There Is Only Software” (unveröffentlicht).

http://www.manovich.net/DOCS/Manovich.there_is_only_software.pdf.


