SOFTWARE STUDIES EXAM 2018

Object Orientation

# Draft\_1:

**Idea/entry/concept:**

Object orientation – The genealogy of the digital object and its possible effect on society

Possible themes:

Language, philosophy, ontology, genealogy of digital object, genealogy of object, history of object orientation the three different languages: Simula, Sketchpad, Smalltalk , two different philosophies about object orientation/new ontology, politics in object orientation, Galloway object orientation way to global capitalism.

**Problem:**

Object orientation as an abstraction of the real world, rise different relevant question in relation to language, society, politics, philosophy, programming etc. My interest in this subject rise from the act of having to include some things and exclude others when trying to make an abstraction of a concept from the real world. How does this act affect our understanding of society today?  
 What I am beginning to understand is language role in this action. And the question of whether or not objects are materials, or conceptualizations created through language, both in real world and in programming, or if they in fact are both.

To structure my paper, I will choose an object/example with which I will try to develop an understanding of language role in regard to objects, but also object orientations effect on the way we view society. I will try to examine whether or not objects can form the language, or if the language forms the object?

Object examples:

* Metrics
* Buttons
* Specific Software: f.ex. word

Inspiration quotes from “The invention of the object – object orientation and the philosophical development of programming languages” (Justin Joque):

I have found a lot of inspiration for my paper from this article, therefore I have included some quotes from the article that touches upon some different themes I find relevant in regard to my paper.

The article describes how objects was invented through history. Which persons, and which programming languages that lead to the understanding of objects and object-oriented programming we have today. To some extent the author tries to examine the philosophy behind object orientation and language role in invention of objects.

Interesting points from the conclusion that I could develop on:

I will read the text, later mentioned, of Galloway. And examine the political structures in object-oriented programming. As I have understood, Galloway argues that the philosophers working with object orientation does not address the problems that can rise trying to describe the world through objects.

“Galloway claims that object-oriented philosophy (and with it a number of philosophies built on mathematical foundations, such as Badiou and Mellasioux’s work) is politically implicated in its structural complicity with object- oriented programming, which he sees as the central technology underwriting global capitalism.” (p.354)

” Despite Galloway’s recognition of this anti-political tendency, it seems clear in the light of the history of object-oriented programming that there is in fact a large gulf between object-oriented programming and object-oriented philosophy.” (p.355)

I want to address the question of whether or not the real can be described and understood through objects, and mostly digital object. I will focus on the digital object to make sure my perspective will be relevant in a software studies article:

”If we under- stand objects to be an abstraction of the real that ease our ability to understand and live within its complexity, all the while obfuscating the underlying processes, it becomes possible to envision a form of object-oriented philosophy that would seek not to understand what objects exist but how and to what ends the real could be described as objects, processes, or other ontological categories.”(ibid)

” Against the solidity of objects gaining popularity in certain quarters, a philosophy informed by the genealogy of the digital object must confront and understand the multitude of ways in which the real can be divided, described, and understood.” (ibid)

**Plan (incl. next steps):**

Step 1: I need to decide on an object I will use as an example throughout the paper. If I choose metrics, it would be very relevant to use the text from class: Benjamin Grosser,” What Do Metrics Want? How Quantification Prescribes Social Interaction on Facebook,”. I would also like to explore the internet, trying to find an object that haven’t already been described.

Step 2: I will read the text: Galloway, A. (2013). The poverty of philosophy: realism and post-Fordism. Critical Inquiry, 39(2), 347–366. Examining whether or not Galloways perspective on the difference between object orientated programming and object orientation philosophy, can be the offspring of my paper. Trying to examine the gulf between, two maybe different ontologies.

Step 3: I should read a wide variety of text to research different approaches to object orientation. Afterwards I will have to decide what my focus will be, either the technical programming aspect and focus on OOP (object-oriented programming), focus on the philosophy of object orientation and its effect on society (how we view the world, politics, object rights), or which role both natural language and programming language play in working with objects. Maybe I can make a thesis that can combine some of these subjects.

Step 4: When I have decided on an object example and the entry to my paper, I will try to structure my paper in different sections, to make sure that the overall paper makes sense both in chronological, but also in relation to the examinations requirements for the software studies exam.

Step 5: Make final synopsis.

**Texts:**

I have chosen some texts from class, that I am sure to use. Apart from that there is a section called possible texts, that I will need to read again to see if there are some relevant perspectives on my thesis, I can use when I begin to write.

I have found some additional texts. Two of them I am sure I’m going to use, the rest of them I still need to read, to see if they are relevant to my subject.

From class:

* Florian Cramer, “Language”, in Fuller op. cit., pp. 168-174.
* Benjamin Grosser,” What Do Metrics Want? How Quantification Prescribes Social Interaction on Facebook,” in *Computational Culture* no. 4 (2014). <http://computationalculture.net/article/what-do-metrics-want>

Also look at Grosser's projects: <https://bengrosser.com/projects/>

* Cecile Crutzen and Erna Kotkamp, "Object Orientation", in Fuller, op. cit., pp. 200-207.

Possible texts:

* Richard Wright, "Data Visualisation", in Fuller, op. cit., pp. 78-87.
* Matthew Fuller, “It Looks Like You’re Writing a Letter: Microsoft Word”, in Behind the Blip!, New York: Autonomedia, 2003; early version available at <http://www.nettime.org/Lists-Archives/nettime-l-0009/msg00040.html>
* Michael Mateas, “Weird Languages”, in Fuller op cit., pp. 267-275 (Daniel Temkin, “Code Art, Code Poetry, and Esolangs”, in Esoteric.codes, 2014, <http://esoteric.codes/post/104079889888/code-art-code-poetry-and-esolangs)>
* Geoff Cox & Alex McLean, “Not Just for Fun”, in Olga Goriunova, ed. Fun and Software, London: Bloomsbury, 2014, pp. 157-173.
* Derek Robinson, "Variable", in Fuller, op. cit., pp. 260-266. & Friedrich Kittler, “Code (or, How You Can Write Something Differently)”, in Fuller, op. cit., pp. 40-47.
* David M. Berry, “The Idea of Code”, in The Philosophy of Software: Code and Mediation in the Digital Age, Palgrave, 2011, pp.1-28.
* Derek Robinson, "Variable", in Fuller, op. cit., pp. 260-266. & Friedrich Kittler, “Code (or, How You Can Write Something Differently)”, in Fuller, op. cit., pp. 40-47.

Additional texts:

* Jaque, Jason. (2016). The invention of the object – Object orientation and the philosophical development of programming languages. Springer Science+Business Media Dordrecht.
* Galloway, A. (2013). The poverty of philosophy: realism and post-Fordism. Critical Inquiry, 39(2), 347–366.

Not read yet(possible texts) :

* Evens, A. (2006). Object-oriented ontology, or programming’s creative fold. Angelaki: Journal of Theoretical Humanities, 11(1), 89–97.
* Harman, G. (2011). On the undermining of objects: grant, Bruno, and radical philosophy. In L. Bryant, N. Srnicek, and G. Harman (Eds.), The speculative turn: continental materialism and realism (pp. 21–40). Victoria: re. Press, 2011.
* Hayles, K. (1999). How we became posthuman: virtual bodies in cybernetics, literature, and informatics. Chicago: University of Chicago Press.
* Kay, A. and Ram, S. (2003). On the meaning of Bobject-oriented programming. ^ E-mail exchange. http:// userpage.fu-berlin.de/~ram/pub/pub\_jf47ht81Ht/doc\_kay\_oop\_en. Accessed: 15 December 2015
* Robson, D. (1981). Object-oriented software systems. Byte Magazine, 6(8), 74–82.
* Smith, B. (1996). On the origin of objects. Cambridge: MIT Press.