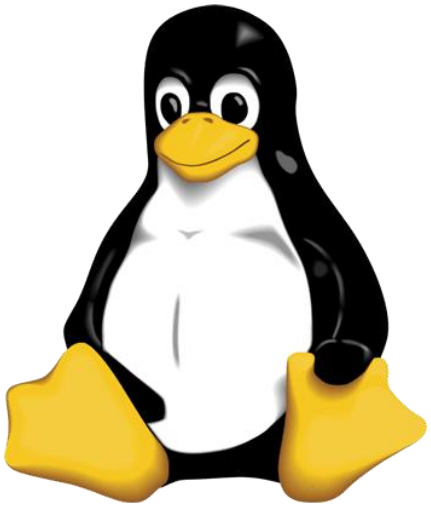
Kaikki huutaa MOOC!

Jaakko Kurhila

SULOP 2013 Helsinki, 7.3.2013



MOOC

massive open online course

Udacity
Coursera
edX

<Osa 1>

hype

New York Times: ”2012 oli MOOCien vuosi”

Web Search Interest: **mooc**. Worldwide, Jan 2007 - Jan 2013.

Interest over time

The number 100 represents the peak search volume

News headlines Forecast

Miksi hype?

Aika on vihdoinkin kypsä: **toimii**

Yhteydet, välineet, toimintakulttuuri

…facebook, youtube, googlaus, wikipedia

…plus joustavuus, vihdoinkin

Yhteydet, välineet, toimintakulttuuri

- facebook, youtube, google

Evidenssiä että ”homma toimii”

- AI-class (Stanford U) 160000 aloittajaa
- Khan Academy videoita katsottu *n* kertaa

“I can see a day soon where you’ll create your own college degree by taking the best online courses from the best professors from around the world — paying only the nominal fee for the certificates of completion. It will change teaching, learning and the pathway to employment.”

- Thomas Friedman in NYTimes 27 Jan 2013

“I can see a day soon where you’ll create your own college degree by taking the best online courses from the best professors from around the world — paying only the nominal fee for the certificates of completion. It will change teaching, learning and the pathway to employment.”

- Thomas Friedman in NYTimes 27 Jan 2013

MOOC on myös **hyvä tarina:**

parasta mahdollista ilmaiseksi kaikille!

Khadija Niazi

Khadija Niazi, 11v tyttö Pakistanista, suoritti Udacityn MOOCin ”Physics100”

Potentiaalinen bisnes: **rahaa** liikkuu

ja paljon

Venture-Capital Investment in Education-Technology Companies

Note: Data include educational-technology companies in elementary and secondary education, higher education, lifelong learning, and informal education.

Source: National Venture Capital Association, thomson Reuters

USAssa riskirahan kasvu jatkunut:
2012 jo
$1.87 miljardia


USAssa riskirahan kasvu jatkunut:
2012 jo
$1.87 miljardia
fir0002 | flagstaffphotos.com.au

Miksi rahaa liikkuu?

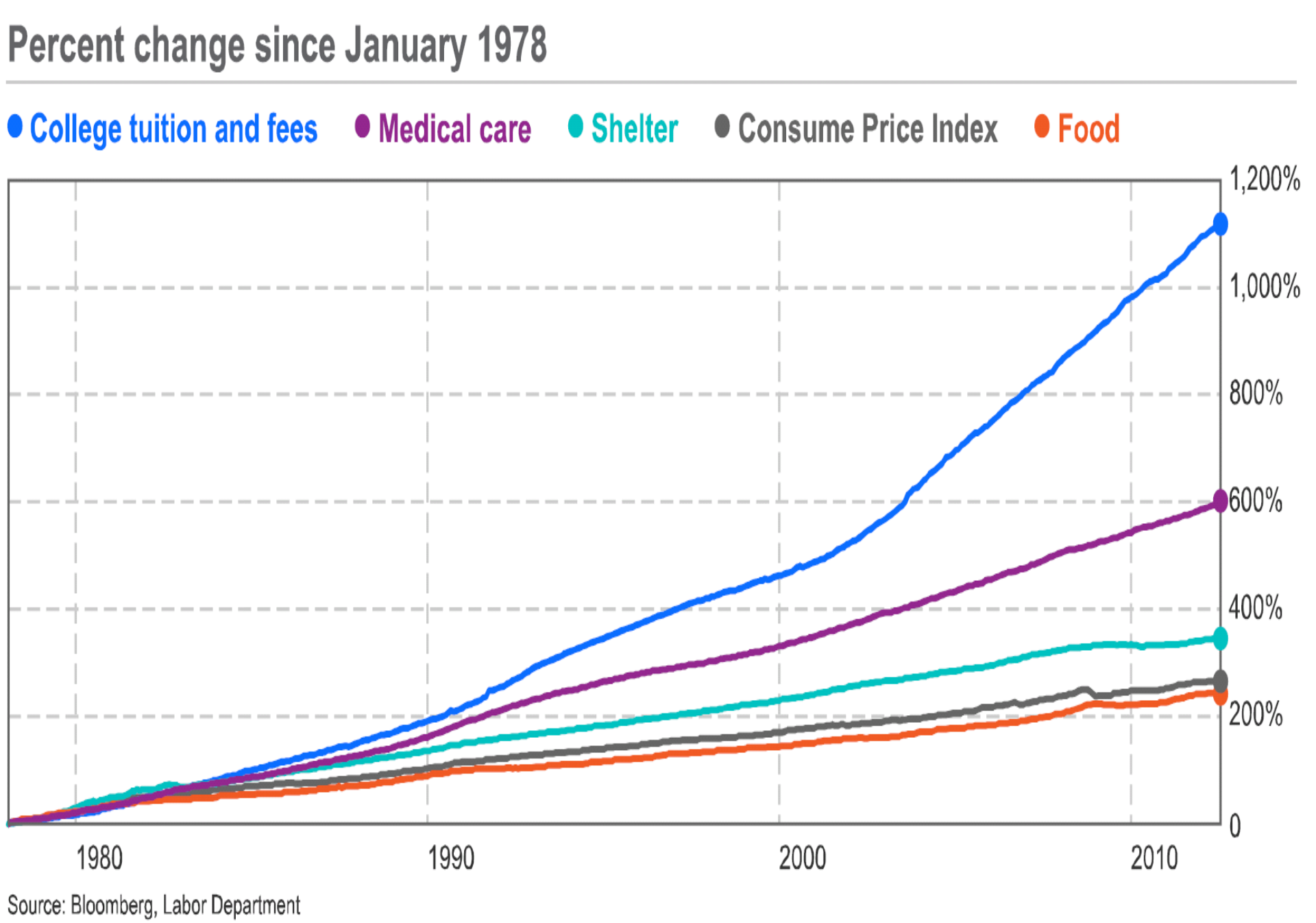
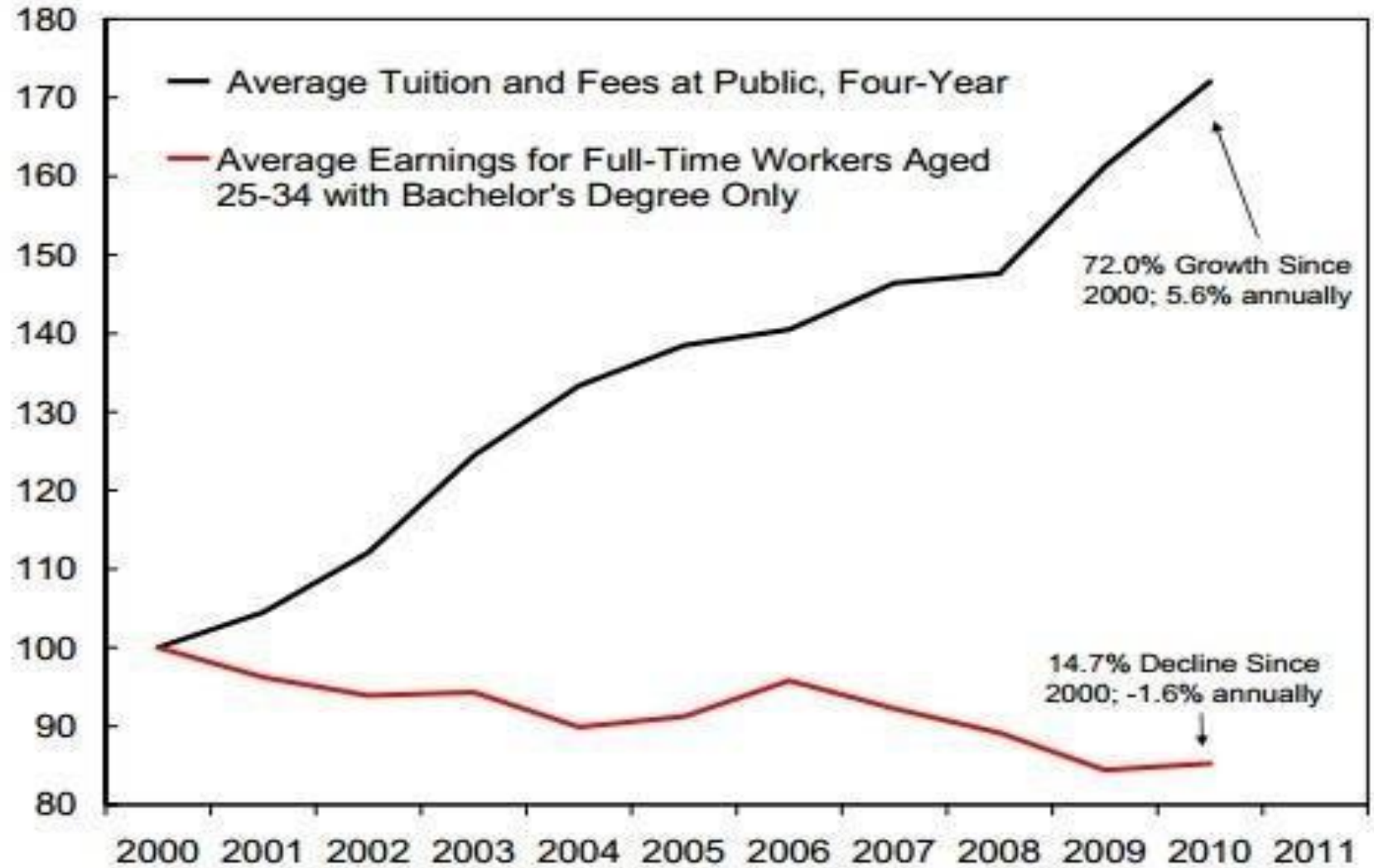

Percent change since January 1978
College tuition and fees
Medical care
Shelter
Consume Price Index
Food
1,200%
1,000%
800%
600%
400%
200%
0
1980
1990
2000
2010
Source: Bloomberg, Labor Department

Figure 5. Real Tuition and Fees at a Public, Four-Year College and Average Earnings for Full-Time Workers Aged 25–34 with Bachelor's Degree Only (Indexed, 2000 = 100)



Sources: College Board, U.S. Department of Education, Census Bureau, and Citi Research. Note: Both tuition and earnings were weighted in 2010 dollars, and tuition and fees were enrollment-weighted.

“The University of California plans to beef up its online classes enough to make them as much as 11% of the undergraduate curriculum over the next five years […] And the university was clear -- this move is not about improving things for students or about moving teaching into the 21st century, either. It's about money, or the lack of it, rather.”

-- San Francisco Business Times, 26 Jan 2013

Miksi koulumaksut kallistuvat?

Baumolin tauti

”tuottavuuden ja palkkojen kasvaessa useilla talouden aloilla myös niillä aloilla, joilla tuottavuuden kasvua ei ole tapahtunut, joudutaan nostamaan palkkoja”


1350

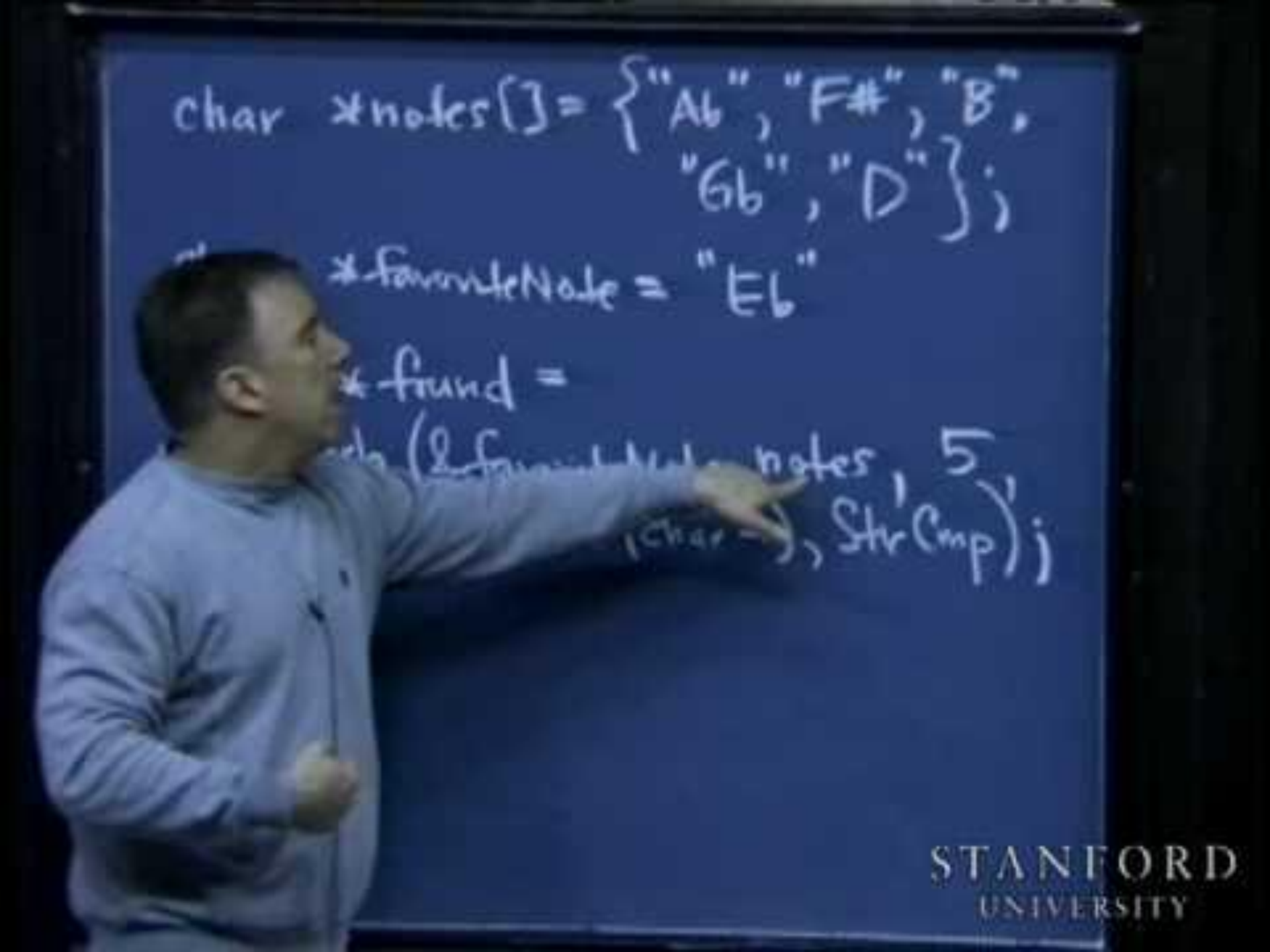

1820

2013

<Osa 2>

MOOC-pedagogiikka

Tämän kaikki maallikotkin *tietävät* yliopistosta:


char *notes[] = {"Ab", "F#", "B",
"Gb", "D"};
*favoriteNote = "Eb"
*found =
notes, 5,
(char -), StrCmp);
STANFORD
UNIVERSITY

Yliopisto-opetuksen ydin on tuhat vuotta ollut ***luento***

Kuvastavatko (nyky)MOOCit sitä, mitä ulkopuoliset luulevat, että yliopistot tekevät?

Nykyiset hype-MOOCit rinnastuvat stadionkeikkoihin:

Opettaja, sinä voit olla *tähti*

Functional Programming Principles in Scala

Martin Odersky

Learn about functional programming, and how it can be effectively combined with object-oriented programming. Gain practice in writing clean functional code, using the Scala programming language.

Next Session:
Mar 25th 2013 (7 weeks long) Sign Up

Workload: 5-7 hours/week

About the Course

This course introduces the cornerstones of functional programming using the Scala programming language. Functional programming has become more and more popular in recent years because it promotes code that's safe, concise, and elegant. Furthermore, functional programming makes it easier to write parallel code for today's and tomorrow's multiprocessors by replacing mutable variables and loops with powerful ways to define and compose functions.

About the Instructor

Artificial Intelligence for Robotics

UDACITY

Advanced

INSTRUCTORS

Sebastian Thrun

Take the Class

Class Summary

Learn how to program all the major systems of a robotic car from the leader of Google and Stanford's autonomous driving teams. This class will teach you basic methods in Artificial Intelligence, including: probabilistic inference, planning and search, localization, tracking and

What Should I Know?

You should either already know Python, or have enough experience with another language to be confident you can pick up what you need on your own. Fortunately, Python was built to be easy to learn, read, and use. If you already know another programming language, you'll be coding in

What Will I Learn?

This course will cover probabilistic inference, planning and search, localization, tracking and control, all with a focus on robotics.

Course Instructors

Sebastian Thrun

Instructor

Introduction to Physics

UDACITY

Beginner

INSTRUCTORS

Andy Brown

Jonathan Burket

Take the Class

Class Summary

Study physics abroad in Europe -- virtually! Learn the basics of physics on location in Italy, the Netherlands and the UK, by answering some of the discipline's major questions from over the last 2000 years.

What Should I Know?

This course is suitable for anyone; a basic understanding of algebra is suggested.

What Will I Learn?

This unique class gives you the chance to see the sites where physics history was made and learn some of the subject's most captivating concepts.

Course Instructors

Andy Brown

Instructor

After graduating from MIT in 2009 spent two years teaching, travelin the world, and biking across the U

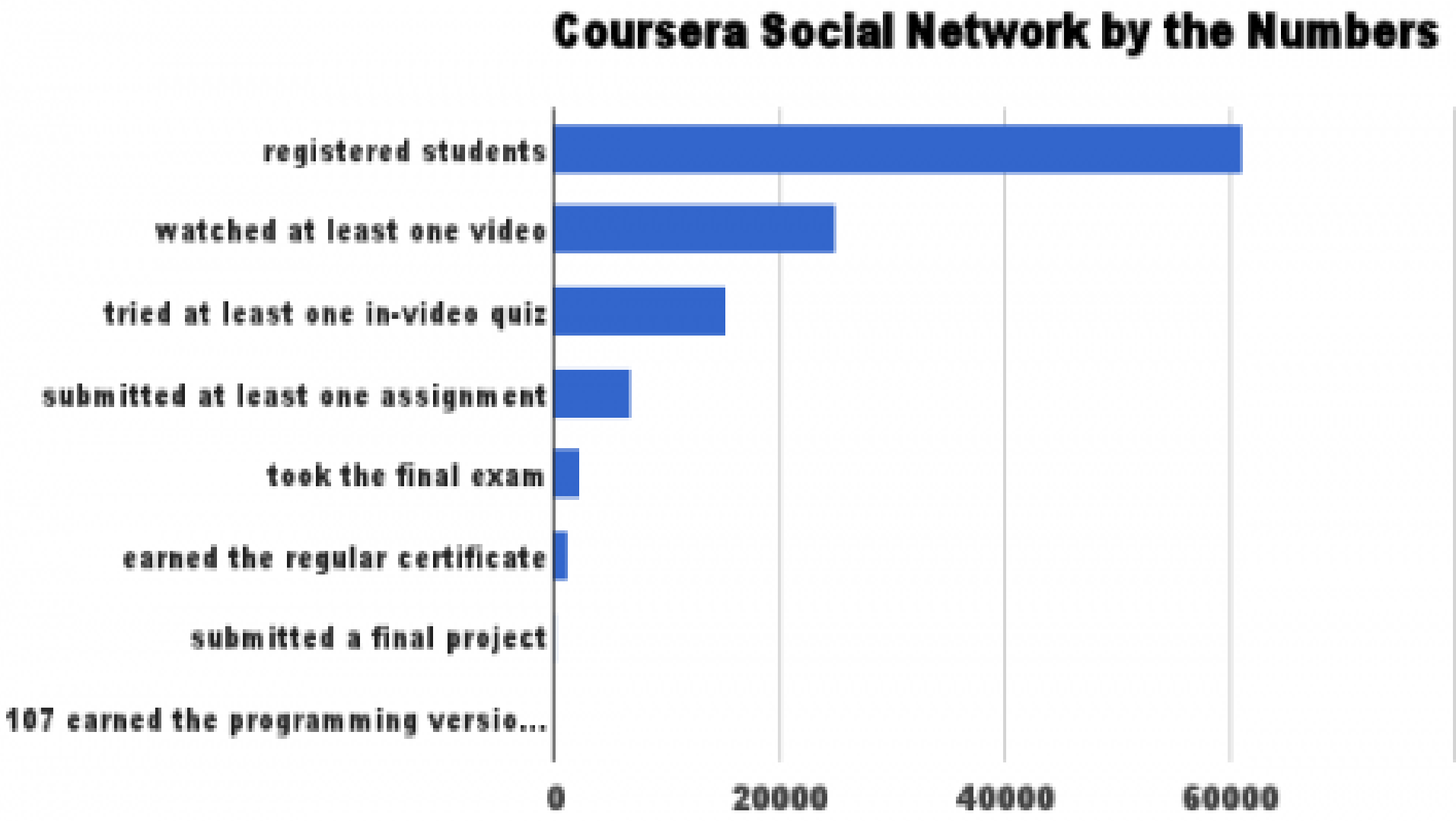
Syllabus

Unit 1: How can we measure the circumference of the Earth?
Basics of geometry and trigonometry

Unit 2: How do objects move?
Data analysis and kinematics

Oppiminen = muutos

- merkittävä henkilökohtainen muutos vaatii paljon



Alan Levine, 27 Nov 2012

Lisäksi: moni on tehnyt saman kurssin osana aiempia opintojaan, ja:

edX: Circuits & Electronics

37% BSc, 28% MSc, 6% PhD

Coursera: Functional Programming

36% BSc, 45% MSc, 6% PhD

Mutta MOOC olikin bisnes!

Wikimedia Commons

Rekrytointi on $130 miljardin bisnes

-- Forbes.com, Dec 2012

Huom! Courseralla 2 miljoonaa käyttäjää nopeammin kuin Facebookilla tai Twitterillä

Riski- tai siemenrahoitusta merkittäville toimijoille:

edX (nonprofit)	$60m
Minerva Project (aloittaa 2015)	$25m
Coursera	$23m
Udacity	$17m

2012 oli kilpajuoksu huipulle (firmoille), harva yliopisto vielä mukana:

- 33 + 29 yliopistoa liittynyt Courseraan
- MIT ja Harvard muodostivat edX:n, johon liittynyt 10 muutakin huippua
- FutureLearn 12+6 yliopistoa (OU + Library)

Mukaan mennään pelosta:
”mitä jos sittenkin jäädään paitsi jostain”

Samalla tilanne elää koko ajan:

MOOC → xOOC → xxOC → xxxC

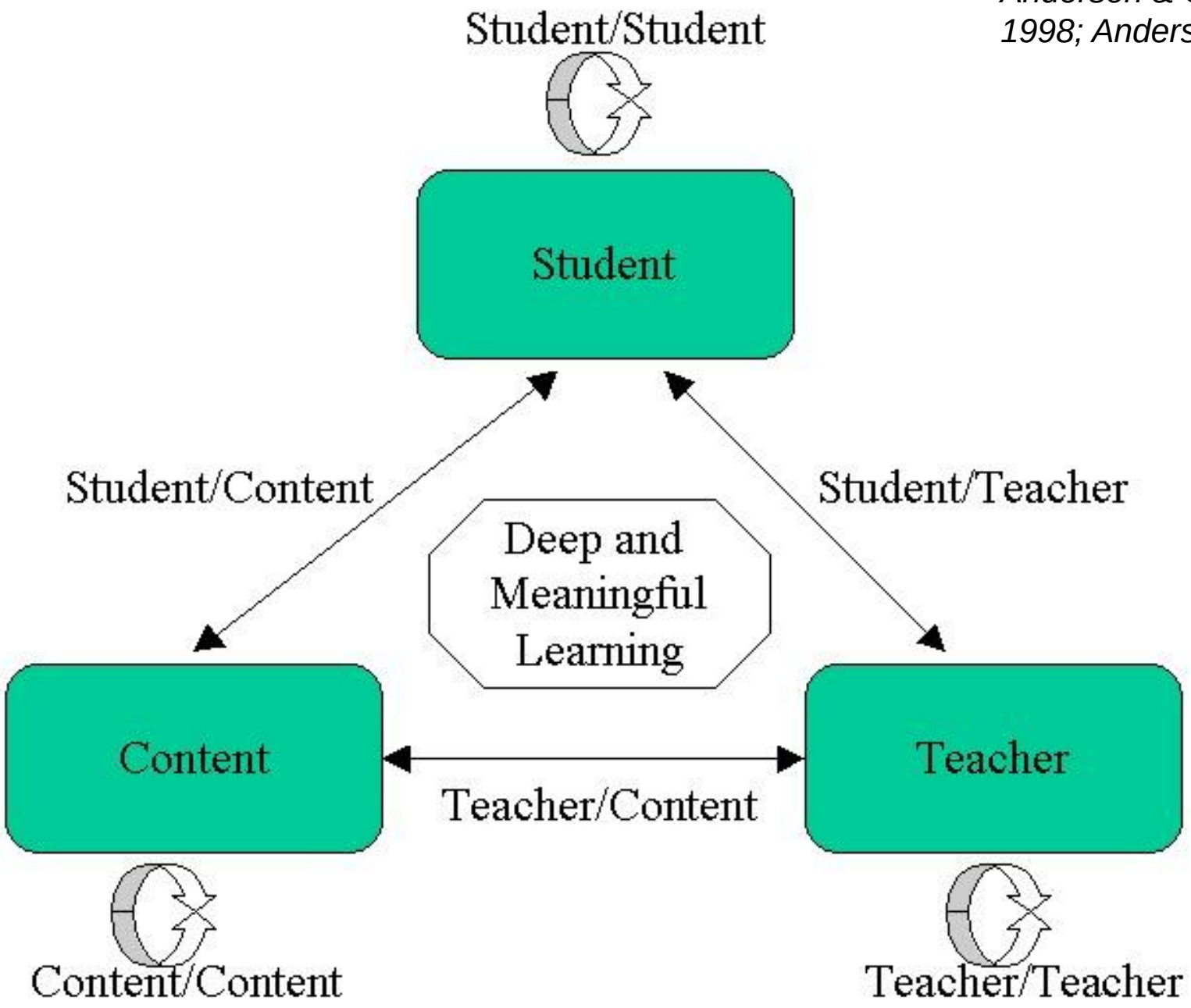
Missä hype-MOOCit menevät pieleen?

Hype-MOOCit toistavat epäoptimaalisia *opettamisen* rakenteita (eivätkä ole web 2.0)

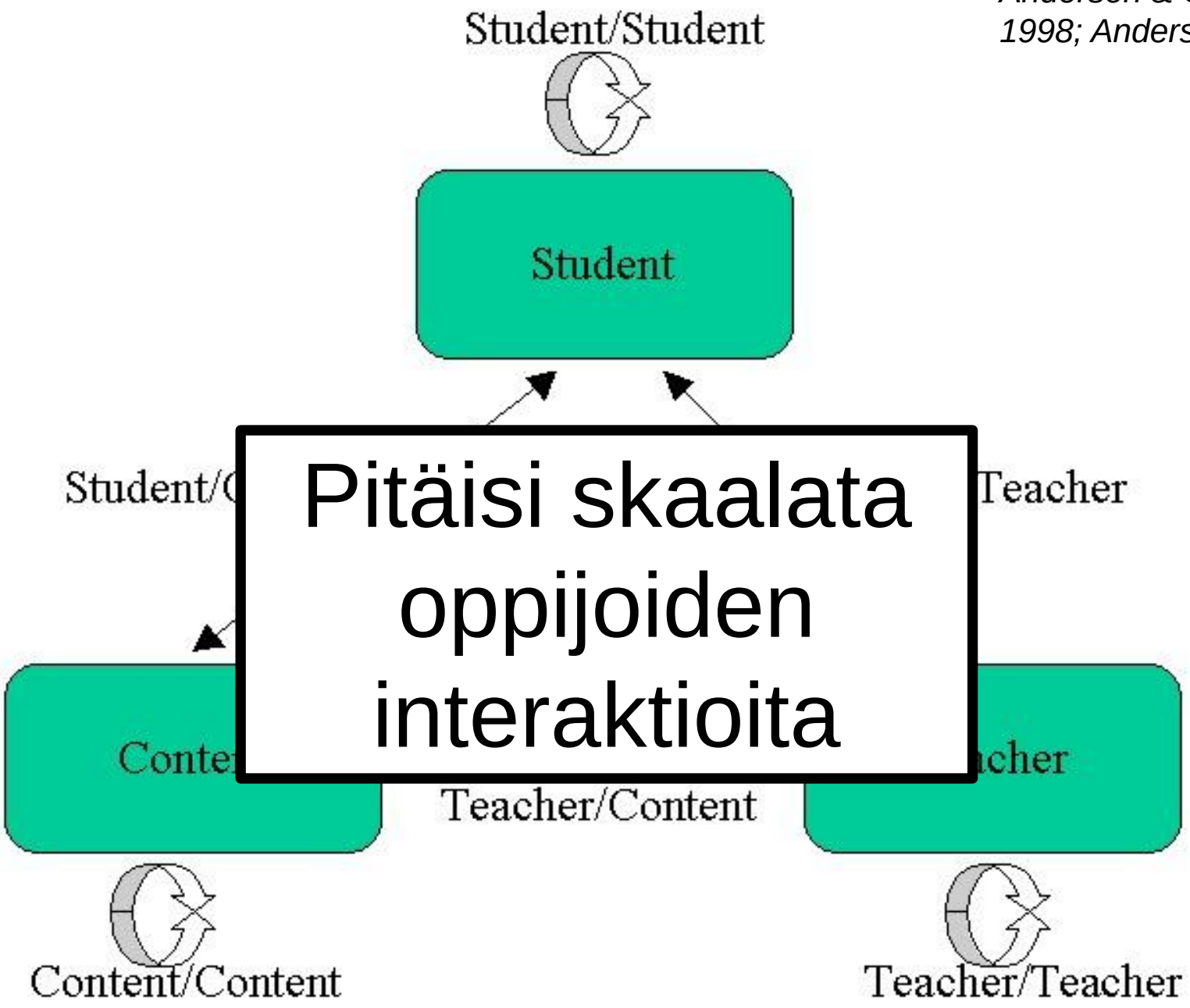
Hype-MOOCit skaalaavat tätä:

hype-moocit skaalaavat väärää asiaa:

Anderson & Garrison 1998; Anderson 2002



Anderson & Garrison 1998; Anderson 2002



Pitäisi skaalata oppijoiden interaktioita

(Toki alkuperäiset cMOOCit tajusivat tämän)

<Osa 3>

Mitä hyötyjä nopeastikin?

learning analytics

learning analytics

suunnilleen sama asia kuin

educational data mining

alueella alustavia askeleita…

Mitataan käytettyä aikaa, virheitä, klikkauksia, polkuja – kun dataa on paljon, näillä alkaa olla merkitystä

Mitataan käytettyä aikaa, virheitä, klikkauksia, polkuja – kun dataa on paljon, näillä alkaa olla merkitystä

periaatteessa voitaisiin oppia oppimisesta

Nyt voidaan identifioida heitä, jotka ovat vaarassa pudota; voidaan tunnistaa osallistuja näppäimistön käyttötavan perusteella

Koska *learning analytics* –näkökulma on kilpailuvaltti, voin kertoa vain omastamme

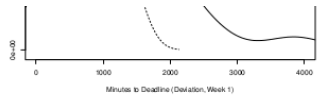


Figure 1: Feature "Minutes to Deadline (Deviation, Week 1)" displayed using probability densities for groups that have passed and failed the course Introduction to

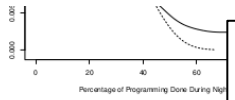


Figure 3: Feature "Percentage o Done During Night (Week 6)" displ ability densities for groups that h

Based on the students' performance in the course exam, the students' have been split into three groups; "fail" (n=36), "pass" (n=53), and "excellent" (n=63). Students in group "fail" failed the course (received less than 50% of the possible score), group "pass" passed the course but got less than 90% of the possible score, and group "excellent" contains students that received 90% or over of the possible score.

During the work, the data was transformed to a form that can be used for classification. Each data item that corresponds to students' programming behaviour is described with almost 200 distinct features, in addition to the label that identifies the group. In order to visualize how features can indicate different groups, let us look at Figure 1, that descibes the feature "Minutes to Deadline (MAX)" (students' furthest working time from the deadline of an exercise).

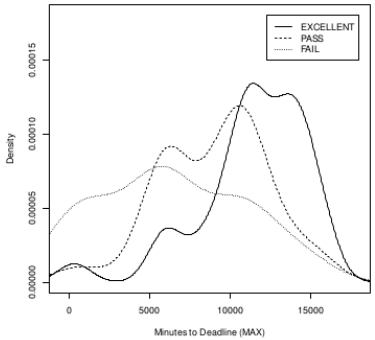


Figure 1. Probability densities for feature "Minutes to Deadline (MAX)" for students in different groups.

The feature "Minutes to Deadline (MAX)" can be interpreted as *eagerness*, and it is a positive factor. Students that start working early on course exercises belong more probably to the group "excellent", while students that start

Figure 2: (Maximu densities course Int

Figure 3 di nights duri between 22 ematics co grammed the mathe

racy. After

Week
1
2
3

be too the we

ca the res

Co

Co

cro tha off we dis pre

[1]

[2]

in *ACM SIGCSE Bulletin*, vol. 36, no. 4. ACM, 2004, pp. 101–104.

[3] S. Wiedenbeck, "Factors affecting the success of non-majors in learning to program," in *Proceedings of the first international*

Predicting Students' Performance in an Introductory Programming Course using Data from Students' own Programming Process

Anonymized
Department of Computer Science
University of City, Country
Email: anony@mou.se

***Abstract*—As the amount of data and the facilities and tools for understanding students' programming process is improving, the time is ripe for analyzing students' actual programming process. In our current work we are investigating how students' behaviour during her programming process (e.g. eagerness to start working on freshly released exercises, following best programming practises) affects the students' results. We purposefully utilize only data gathered automatically using snapshots from the students' programming process, and do not gather any additional background information. Currently, we are able to predict whether the student is a high-performer, passes the course, or fails the course with a 78% accuracy.**

***Keywords*-Computer Science Education, Automated Assessment Service, Extreme Apprenticeship, Code Snapshots, Student Performance**

I. Introduction

Although programming has been a hot topic in education for decades, teachers and students' alike still tend to disregard the meaning of good working habits and programming practises such as Clean Code [1]. In our work, we have been evaluating students' programming process and seeking indicators that could be used to predict high-performance or failure in introductory programming courses, with the goal of introducing them back to the students' in the future.

Identifying factors that can be used to predict failure or success in a programming course is not a novel idea, see e.g. [2], [3]. However, to our knowledge, predicting students' performance based solely on working behaviour from actual programming process has not received much attention. With the rise of massively open online courses ("MOOCs"), where students' can sign up simply by using an alias, the amount of data as well as the need for indicators

II. Course Context and Tools

We organize our programming courses using the Extreme Apprenticeship (XA) method [4]. Most of the work and learning happens in aided computer labs, where the students work on assignments under scaffolding from more experienced programmers, e.g. course instructors and peers.

The students work with industry-strength tools from day one. Our choise is the NetBeans-environment with an automated assessment service called Test My Code (TMC). TMC contains a NetBeans plugin that automatically downloads the exercises to the students' machine, has a capability for gathering snapshots from the students' programming process, and a backend-server for assessing students' work.

As students sign up for a programming course, the are prompted for user details and asked whether they want to send snapshots of their progress of study (*"yes"* by default, can be switched off at any time). If students allow sending snapshots, the system records a snapshot of their current code every time the student saves the exercise.

III. Data and Features

We gathered over 66 000 snapshots from 152 participants of our six-week introductory programming course in Java. As we are interested in actual working practises and their effect on the programming course results, we have purposefully left out "obvious" features such as the amount of completed exercises so far, as different institutions give different amounts of points from the exercises.

Currently we describe the students' working behaviour using features that contain weekly and total data from remaining time to deadline for the snapshot, time between snapshots, and edit distance between snapshots. For each of

MOOC.fi

AVOIMET MASSIIVISET VERKKOKURSSIT

"tää on kyl syvällisin ohjelmointikurssi mitä mä oon ikin käyny ... aivan armotonta"

"Täytyy sanoa, että MOOC:in jälkeen alkaa ymmärtää miksi olio-ohjelmointi on kehitetty"

"nyt on sellanen fiilis et duunis hanskat tiskiin välittömästi ja koulun penkille takas"

Kurssit:

OHJELMOINNIN MOOC

Ohjelmointi on tietokoneen käskemistä. Kurssilla perehdytään algoritmien laatimiseen ja nykyaikaisen ohjelmoinnin perusideoihin. Opiskelijalta ei edellytetä ennakkotietoja ohjelmoinnista. Tämä on se MOOC, joka liittyy opintooikeuden saamiseen Helsingin yliopistoon. **Kurssi on alkanut ja ensimmäinen deadline on 24.2.2013 klo 23.59.**

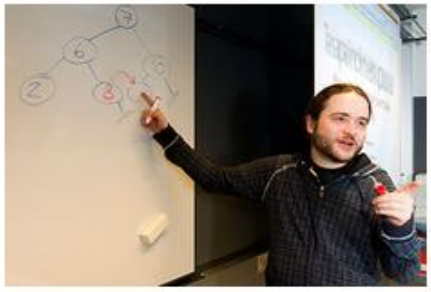
Mene kurssille »

PELIOHJELMOINNIN MOOC

Peliohjelmointikurssilla tutustutaan peliohjelmoinnin perusteisiin käyttäen Java-ohjelmointikieltä. Kurssin tavoitteena on tutustua ohjelmoinnin perusteisiin ja innostaa jatkamaan harrastusta. **Kurssi on alkanut, eikä siinä ole deadlineja.**

Mene kurssille »

ALGORITMIEN MOOC



Algoritmit, eli "mekaaniset toimintaohjeet ongelmien ratkaisemiseen", ovat oleellinen osa ohjelmointia ja tietojenkäsittelytiedettä. AlgoMOOC tarjoaa lisähaastetta Ohjelmoinnin MOOCin suorittajille ja muille kiinnostuneille. Tehtävien ratkaisemisessa ei enää harjoitella pelkkää ohjelmointia, vaan ohjelmoimalla ratkaistavia ongelmia. **Kurssi on alkanut 11.2.2013.**

Mene kurssille »

PROGRAMMING MOOC

In the Programming MOOC participants learn basics in object-

WEB-PALVELINOHJELMOINTI

Kurssilla tutustutaan nykyaikaisten, skaalautuvien ja virheenkestävien web-sovellusten toteuttamiseen, testaamiseen ja julkaisemiseen. Kurssilla keskitytään

Twitter

mooc
mooc1

linjaaho Sano MOOC kun haluat opetusta online bit.ly/UCA7pi #mooc Hyvä Jaakko ja kumppanit!
4 hours ago · reply · retweet · favorite

mooc1 @linjaaho Danke!
4 hours ago · reply · retweet · favorite

STTInfo MOOC- eli avoin online-opetus kasvattaa suosiotaan myös Helsingin yliopistossa: Jos MOOC, englannin massive open... bit.ly/137sWF1
7 hours ago · reply · retweet · favorite

rageresearch Jaakko will give a talk about MOOCs at Free University of Bozen-Bolzano, Italy, on Wed. You can follow the talk at inf.unibz.it/stream/
4 hours ago · reply · retweet · favorite

mooc1 Ohjelmoinnin MOOCn viikon 5 tehtävät on julkaistu! Vielä ehtii mukaan, sillä viikon 1 deadline on vasta 24.2. mooc.fi
5 days ago · reply · retweet · favorite

LUMAsanomat Opiskelijaksi yliopistoon? blogs.helsinki.fi/wdc-2012/2013/...
7 days ago · reply · retweet · favorite

mooc1 SpeedRun Challenge:n tulokset: 1. Kultaa: Metabolix ajalla 3h 21m 2. Hopeaa: poiru ajalla 5h 19m 3. Pronssia: Aleksi Salmela ajalla 14h 20m
7 days ago · reply · retweet · favorite

mooc1 @frbbs Pahoitteluni, en ollut huomannut viestiäsi. Ohjeet ongelman poistamiseen on lisätty sivulle mooc.fi/ohjelmointi/oh... kohtaan 10
7 days ago · reply · retweet · favorite

mooc1 Hopeaa: poiru ajalla 5:19 tmc.mooc.fi/mooc/courses/1...
7 days ago · reply · retweet · favorite

mooc1 Algoritmien MOOCin SpeedRun Challengen voitti Metabolix ajalla 3:22! Onnittelut!
7 days ago · reply · retweet · favorite

Join the conversation

Ohjelmoinnin opiskelua ajatuksena:

asian *katseleminen*
on eri kuin
asian *tekeminen*

Netbeans with TMC 0.3.14

File Edit View Navigate Source Refactor Run Debug Team Tools TMC Window Help

Search (Ctrl+I)

<default config>

Projects Files Services

- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries

JavaApplication1.java - Navigator

Members View

- JavaApplication1
  - main(String[] args)

JavaApplication1 Run tests locally

Source History

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package javaapplication1;
6
7  /**
8   *
9   * @author kurhila
10  */
11 public class JavaApplication1 {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19 }
20
```

1 | 1 INS

Netbeans with TMC 0.3.14

File Edit View Navigate Source Refactor Run Debug Team Tools TMC Window Help

Search (Ctrl+I)

<default config>

Projects Files Services

- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries

JavaApplication1.java - Navigator

Members View

- JavaApplication1
  - main(String[] args)

JavaApplication1 Run tests locally

Source History

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package javaapplication1;
6
7  /**
8   *
9   * @author kurhila
10  */
11 public class JavaApplication1 {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19 }
20
```

1 | 1 INS

Extreme Apprenticeship (XA)

eli

”tehostettu kisällioppiminen”

XA periaatteet

1. ***start with exercises***; use small **incremental** exercises that ensure achievable tasks; exercises need to provide clear guidelines on **how to start** solving the task and when a task is considered **finished**
2. ***exercises define lecture form and content***; minimize lecturing and maximize number of exercises
3. advisor must be **present** in a same space when student is working on the exercises
4. best up-to-date programming practices are emphasized throughout the scaffolding phase
5. students are encouraged to extend their knowledge beyond the instruction provided

XA periaatteet

1. ***start with exercises***; use small **incremental** exercises that ensure achievable tasks; exercises need to provide clear guidelines on **how to start** solving the task and when a task is considered **finished**
2. ***exercises define lecture form and content***; minimize

Zero lectures, spot-on material with *doing* orientation

3. advisor must be **present** in a same space when student is working on the exercises
4. best up-to-date programming practices are emphasized throughout the scaffolding phase
5. students are encouraged to extend their knowledge beyond the instruction provided

XA periaatteet

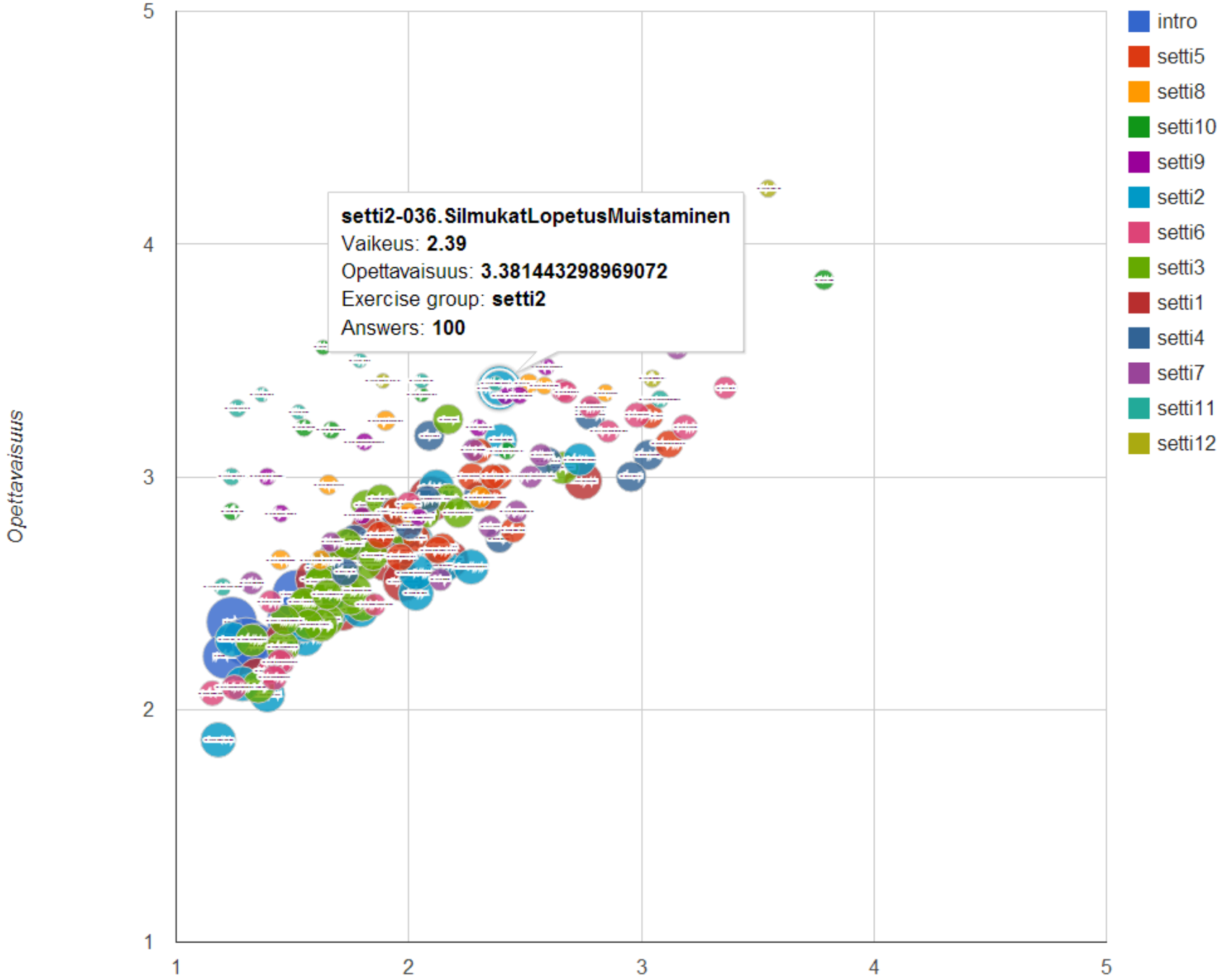
1. ***start with exercises***; use small **incremental** exercises that ensure achievable tasks; exercises need to provide clear guidelines on **how to start** solving the task and when a task is considered **finished**
2. ***exercises define lecture form and content***; minimize

Zero lectures, spot-on material with *doing* orientation

3. adviser must be **present** in a same space when student

Automated tests and chat-based peer-scaffolding

4. best up-to-date programming practices are emphasized throughout the scaffolding phase
5. students are encouraged to extend their knowledge beyond the instruction provided


setti2-036.SilmukatLopetusMuistaminen
Vaikeus: 2.39
Opettavaisuus: 3.381443298969072
Exercise group: setti2
Answers: 100
Opettavaisuus
1
2
3
4
5
intro
setti5
setti8
setti10
setti9
setti2
setti6
setti3
setti1
setti4
setti7
setti11
setti12

Ohjelmointi-MOOCimme tuli ”ilmaiseksi”, sivutuotteena kurssimme kehittämisessä

Paras asia:

MOOC.fi on tiukasti sidottu rakenteisiimme

- omat opiskelijamme toimivat betatestaajina
- MOOC toimii pääsykokeena:
    * kenellä on kykyä ja halua alalle
    * jos kyllä, lentävä lähtö

Learning analytics –tuloksia, esim:

- Pystymme kolmen viikon ohjelmoinnin jälkeen ennustamaan noin 100% varmuudella opiskelijan menestymisen ainoalla pakollisella matematiikan kurssilla (pelkästään tarkastelemalla koneellisesti opiskelijan tapaa ohjelmoida, siis ilman ohjelmointitehtävien suoritustilastoja)

<lopuksi>

NykyMOOCit ovat “oppikirjoja”, joissa itseopiskeluun kykenevät pärjäävät

- ainahan jotkut ovat pystyneet oppimaan, olosuhteista huolimatta

- Fiksut opiskelijamme jo tekevät MOOCeja, halusimme sitä tai emme
  - esim. Codeschool, 2 tunnin täsmäkursseja ajankohtaisista tekniikoista, joista suoraan hyötyä työllistymisessä **vahvan pohjan päällä**

- Fiksut opettajatkin käyttävät MOOCeja
  - Opiskelijoiden kanssa esim. *flipped classroom* –tyyppisesti
  - … plus tietysti oman osaamisen kartuttamiseen

Ei silti kannata ”myydä” yliopistoa halvalla bisnesbrändille – siis yliopistoa yhteisönä, tilana, tilaisuutena

- yliopistoilla on vielä elintilaa itsenäisinä toimijoina eikä Courseran ”alihankkijoina”

Pitääkö olla kiitollinen siitä, että opettamisesta vihdoinkin puhutaan yliopistoissa ja yliopistojen yhteydessä?

Kyllä, kunhan puhutaan *oppimisesta* ja opettamisesta *oppimisen tukemisena*

Kiitos!

http://www.cs.helsinki.fi/rage

http://mooc.fi