

Laboratory of Computer Science Education

Introduzione

Alberto Montresor

Università di Trento

2023/02/27

This work is licensed under a Creative Commons
Attribution-ShareAlike 4.0 International License.



Sommario

- 1 Introduzione
- 2 Pensiero computazionale
 - I fondatori
 - Le iniziative recenti
 - Le iniziative italiane
- 3 Organizzazione a breve

Subito i link!

Sito	cricca.disi.unitn.it/montresor/teaching/lcse/
Questionario	https://forms.gle/agsgysicUbotyJ796
Telegram	https://t.me/+CMM0M2yD40o5Njdk

Riflessioni sul concetto di lezione universitaria



Laurentius da Voltolina – Bologna, seconda metà del XIV secolo

Qualche parola su di voi...

- Perché siete qui?
 - Solo per curiosità
 - Vorreste lavorare in industria ma non si sa mai
 - Il vostro sogno è fare l'insegnante
 - Siete già insegnanti
- Corso di laurea?
- 24 crediti?

Cosa vi aspettate dal corso?

Qualche parola su di me....

- Cosa mi spinge?
- Quali sono i miei obiettivi?
- Due punti di vista (in contrasto? io non credo!)

Those who can, do.
Those who understand, teach

Lee Shulman, 1986

Creativity: Learning creative learning

Quando avete tempo, date un'occhiata a questi due video:

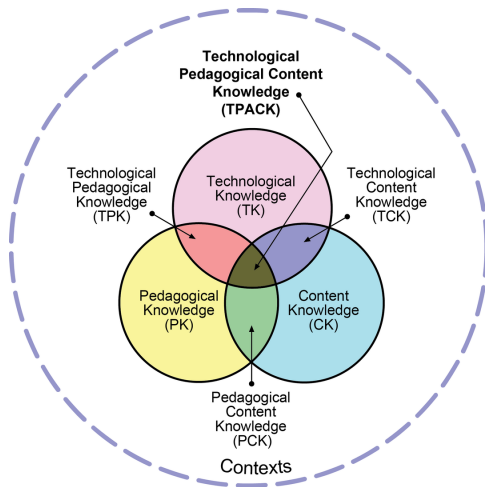
https://www.ted.com/talks/sir_ken_robinson_how_to_escape_education_s_death_valley

https://www.youtube.com/watch?v=A_0XzM34_Ew&ab_channel=SeriousScience

Se interessati ad approfondimenti:

- Mitch Resnick. Lifelong kindergarten: cultivating creativity through projects, passion, peers and play. MIT Press, 2017.
- Ken Robinson, Lou Aronica. Creative Schools: The Grassroots Revolution That's Transforming Education. Penguin, 2018.

(Technological) Content Pedagogical Knowledge



https://en.wikipedia.org/wiki/Technological_pedagogical_content_knowledge

Questo corso è così meta...

Seymour Papert

You can't think about thinking without thinking about thinking about something

Alberto Montresor

You can't teach about teaching without teaching about thinking about teaching



Prerequisiti

What (Else) Should CS Educators Know?

We must first state in the strongest possible terms that a CS educator must have a thorough background in CS, on a fitting level. [...] the fact that high school-level teachers must have a Master's degree in CS is not always sufficiently acknowledged. In fact, this requirement is rarely met; in many high schools worldwide, CS is taught by people who don't even have full undergraduate training in the subject they teach. This situation must change [...] we proceed on the assumption the educator has mastered the field itself as befitting the level being taught. (Gal-Ezer and Harel)

In questo corso, non faremo questa assunzione – ma parliamone!

Organizzazione generale

- **Parte 1 — Fondamenti storico / epistemologici dell'informatica**
 - Informatica come strumento vs. informatica come disciplina.
 - La natura dell'informatica: prospettive matematiche, scientifiche e ingegneristiche.
 - Pensiero computazionale
- **Parte 2 — Aspetti pedagogici**
 - Costruttivismo e costruzionismo
 - Peer instruction, pair programming
 - Didattica problem-based, inquiry-based, challenge-based

Organizzazione generale

- **Parte 3 — Natura e insegnamento della programmazione**
 - Notional machine
 - Misconceptions
 - Ruolo delle variabili e pattern elementari
 - Metodo PRIMM
- **Parte 4 — Aspetti tecnologici**
 - Ambienti e strumenti per l'apprendimento dell'informatica
 - Coding unplugged, programmazione a blocchi, debugging
- **Parte 5 — Argomenti di ricerca**
 - Computing Education Research — stato dell'arte

Una nota su computing education research

- Ricerca "Marco Polo"
- Non è il solito tipo di ricerca a cui siamo abituati; bisogna farci l'abitudine
 - Problemi di riproducibilità
 - Bias potenti

Valutazione

- Cosa posso valutare, se non l'insegnamento?
- Preparazione di attività didattiche da presentare a tutti, durante il corso (brevi)
- Preparazione di attività didattiche "rilevanti":
 - Attività per le scuole
 - Attività per Coderdojo
 - Attività per il Fablab
 - Attività per i vostri corsi (se siete insegnanti)

Sommario

- 1 Introduzione
- 2 Pensiero computazionale
 - I fondatori
 - Le iniziative recenti
 - Le iniziative italiane
- 3 Organizzazione a breve

Pensiero computazionale – Una prima definizione

Wing, Jeannette (2014). "Computational Thinking Benefits Society". 40th Anniversary Blog of Social Issues in Computing. [\[Link\]](#)

Definizione [Wing'14]

- "Thinking as a computer scientist"
- Computational thinking is the **thought processes** involved in **formulating a problem** and **expressing its solution(s)** in such a way that a **computer** – human or machine – can **effectively carry out**.

Nota etimologica:

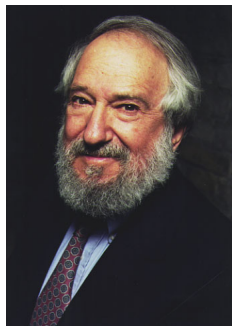
Il termine computer entra nella lingua Inglese negli anni '40... **del 1600!**

Seymour Papert – Anni '80

Il termine "computational thinking" viene introdotto da Seymour Papert

Seymour Papert (1928-2016)

- Ha lavorato con Piaget
- Ha sviluppato la teoria del **costruzionismo**, come evoluzione del **costruttivismo**
- Inventore del linguaggio LOGO



- Papert, Seymour. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc. (1980) [[Link](#)]

Costruzionismo

Costruttivismo

- afferma che l'individuo che apprende costruisce **modelli mentali** per comprendere il mondo intorno a lui
- vede l'apprendimento come una **ricostruzione** piuttosto che come **trasmissione di conoscenze**

Costruzionismo

- sostiene che l'apprendimento avviene in modo più efficiente se questi modelli mentali vengono riflessi nella produzione di oggetti "**tangibili**" e "**significativi**"

Eight Big Ideas Behind the Constructionist Learning Lab

- Learning by doing
- Technology as building material
- Hard fun
- Learning to learn
- Taking time - the proper time for the job
- You can't get it right without getting it wrong
- Do onto ourselves what we do onto our students
- We are entering a digital world - 1999

<http://stager.org/articles/8bigideas.pdf>

Mitchel Resnick – Anni '90

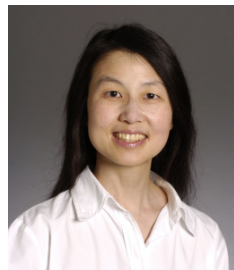
- LEGO Papert Professor of Learning Research
- Ha fondato, insieme a Natalie Rusk, il primo Computer ClubHouse (1993)
- Con il suo gruppo, ha contribuito allo sviluppo dei Lego Mindstorms (1998) e ha sviluppato il linguaggio Scratch (2005)



- Yasmin B. Kafai, Mitchel Resnick, eds. *Constructionism in Practice: Designing, Thinking, and Learning in a Digital World*. Routledge. (1996)

Jeannette Wing – Anni '00

- Corporate Vice President of Microsoft Research
- Ha popolarizzato il concetto di Computational Thinking
- Ha fornito una prima definizione operativa del termine



- Wing, Jeanette M. "Computational thinking". *Communications of the ACM*, 49(3):33, 2006 [[Link](#)]
- Wing, Jeannette. "Computational Thinking Benefits Society". *40th Anniversary Blog of Social Issues in Computing*. (2014) [[Link](#)]

Code.org – 2013

Code.org è un'organizzazione no-profit il cui scopo è incoraggiare le persone, ed in particolare gli studenti di scuole superiori negli Stati Uniti, a imparare l'informatica.



- Organizzatori dell'Ora del codice e della Codeweek negli Stati Uniti
- Supportati da aziende come Microsoft, Apple, Facebook, Twitter
 - Video con Bill Gates, Mark Zuckerberg, Will.I.Am, etc.
- Endorsement da parte del presidente Obama (2015)
- Lobbying per lo sviluppo di nuovi curricula per tutti gli ordini scolastici
- Integrazione razziale, riduzione del gap di genere

Un grosso impatto sui media

The New York Times

U.S.

Reading, Writing, Arithmetic, and Lately, Coding

By MATT RICHEL MAY 10, 2014

Riforme dei curricula scolastici

- Inghilterra (2014-2015)
- Australia, Francia, Polonia, Finlandia (2016-2017)
- Stati Uniti (2017-2018)

Computer Science for All

In the coming years, we should build on that progress, by ... offering every student the hands-on computer science and math classes that make them job-ready on day one.

President Obama
State of the Union Address, 2016

UK National Curriculum in Computing (2014-2015)

Riforma del curriculum scolastico

- Sostituzione della materia "Information and Communications Technology (ICT)" con "Computing", che prevede un'introduzione alla programmazione
- Prevede corsi di aggiornamento degli insegnanti sulle nuove tecnologie

Valutazione preliminare (2 anni)

- Grande successo fra gli studenti
- Solo 15% dei docenti a loro agio con il nuovo curriculum
- Oltre il 60% dei docenti si auto-valutano non sufficientemente preparati per questa sfida

UK National Curriculum in Computing (2014-2015)

Goals for all pupils

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology

<https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study>

Pensiero computazionale in Italia

PNSD

- **Azione #17:** Portare il pensiero logico-computazionale a tutta la scuola primaria
- **Azione #18:** Aggiornare il curriculum di Tecnologia alla scuola secondaria di primo grado
- **Azione #15:** Sviluppo di competenze digitali applicate

Riforma Gelmini

Risultati di apprendimento comuni a tutti i percorsi liceali:
"Essere in grado di utilizzare criticamente strumenti informatici e telematici nelle attività di studio e di approfondimento; comprendere la valenza metodologica dell'informatica nella formalizzazione e modellizzazione dei processi complessi e nell'individuazione di procedimenti risolutivi."

Programma il futuro – 2014

Programma il futuro è un'iniziativa ministeriale che ha l'obiettivo di fornire alle scuole una serie di strumenti semplici, divertenti e facilmente accessibili per formare gli studenti ai concetti di base dell'informatica.



- Organizzatori dell'Ora del codice in Italia
- Il sito è l'emanazione dell'Azione #17 del PNSD

Gli strumenti disponibili sono [...] progettati e realizzati in modo da renderli utilizzabili in classe da parte di insegnanti di qualunque materia. Non è necessaria alcuna particolare abilità tecnica né alcuna preparazione scientifica

Sommario

- 1 Introduzione
- 2 Pensiero computazionale
 - I fondatori
 - Le iniziative recenti
 - Le iniziative italiane
- 3 Organizzazione a breve

Prossime lezioni

- Lunedì 6/3: "Natura dell'informatica"
- Lunedì 16/3: Computational Thinking

Lecture - Per questa volta

- Judith Gal-Ezer and David Harel. What (Else) Should CS Educators Know? Communications of the ACM 41(9):77-84. (1998) [[Link](#)]