Leveraging online courses to increase student success in a Computer Science degree

Linda Marshall
Addressing the problem of underprepared students in CS1

Look at secondary school and undergraduate skills/outcomes requirements

Overview of online courses

Propose online courses to help students become more prepared for CS1
Typical content of a secondary school CS curriculum:

- using basic software
- searching for information on the internet
- programming
- learning to solve complex problems

Peter Hubwieser, 2012
Computer Science Education in Secondary Schools - The Introduction of a New Compulsory Subject
ACM Transactions on Computing Education
using basic software; and
searching for information on the internet
... can be seen as softer ICT related content.
programming; and
learning to solve complex problems
... are seen as more difficult and relate to CS content.
Two secondary school curriculum specifications:

- Computing At School School (CAS) curriculum from the UK
- K-12 Computer Science Standards (CSTA K-12) which is developed by CSTA and ACM in the USA
CAS sees CS as a STEM discipline, characterised by:

- **S** - following a scientific approach
- **T** - understanding, appreciating and applying many technologies to a problem
- **E** - following a process for the construction of artifacts in the discipline, specifically the design-construct-test cycle
- **M** - a mathematical foundation
CAS skills outcome - *computational thinking*
Some clarification is required....
The learner needs to be able to *recognise* computational aspects in the world, *apply* tools and techniques to the recognised systems, and then *understand* and *reason* about these systems. In order to do this, the learner must be able to *abstract*, both by *decomposition* and *generalization*, and *model* the systems. This forms part of the *design* of the system, which then needs to be constructed by programming it before it is *tested*. Understanding the construction of the system will require *fundamental programming*, *algorithm* and *data manipulation* skills. CAS also requires that learners have a basic understanding of *computer architecture* as well as the *internet*. 
CSTA K-12

Characterises a curriculum using strands, these strands define the outcomes of the curriculum.
CSTA K-12

These strands are:

- computational thinking;
- collaboration;
- computing practice and programming;
- computers and communication devices; and
- community, global and ethical impacts
After successful completion of a secondary school curriculum in Computer Science, a learner should:
Secondary school CS outcome S1

S1 have a thorough understanding of theoretical fundamentals of Computer Science which includes algorithms, communication channels such as the internet, data manipulation;
S2 be able to recognise computational problems and then analyse, model, develop and test a computational solution for the problem;
Secondary school CS outcome S3

S3 be able to work with other learners in order to solve a problem; and
Secondary school CS outcome S4

S4 understand the implications of computers on society.
Skill-set required in an undergraduate curriculum as specified in the ACM/IEEE proposed CS2013 Strawman curriculum.
After successful completion of an undergraduate degree programme, a student should have:
Undergraduate CS outcomes G1/2

G1 an in-depth knowledge of topics in Computer Science;

G2 the ability to apply Computer Science in a project environment;
Undergraduate CS outcome G3/4

**G3**  the ability to solve problems on multiple levels of abstraction;

**G4**  organisational and communication skills;
Undergraduate CS outcome G5/6

G5 an understanding that Computer Science is a dynamic discipline and be able to change; and

G6 the ability to interact with other domains.
Introduction
Requirements for Computer Science
University entrance requirements for CS
Online courses
Course selection
Conclusion

Secondary school
Undergraduate
Relationship between skill-sets

Secondary School Skills
- S1
  - Fundamentals of CS
    -> In-depth CS knowledge
  - Analyse, model, develop, test problems
    -> Application of CS to a large project
  - Analyse, model, develop, test problems
    -> Levels of abstraction problem solving

Undergraduate Skills
- G1
  - Group-work to solve a problem
  - Implications of computers and society
    -> In-depth CS knowledge

- G2
  - Application of CS to a large project

- G3
  - Application of CS to a large project

- G4
  - In-depth CS knowledge

- G5
  - In-depth CS knowledge

- G6
  - In-depth CS knowledge

CSERC 2013
Online courses for student success in CS
University admission control mechanisms include:

- using the results of the secondary school to gauge ability and specify relevant admission criteria
- requiring students to write a standard admissions/placement/credit test
- conducting interviews
There is evidence that prior learning has a marked effect on pass rates in first year CS. A study by Morrison and Newman [2001] showed that:

- 66% of first year students with prior learning pass CS 1 with at least a C-grade
- only 50% without prior learning pass CS1 with a C-grade.
## Course providers

### Classification of introductory CS courses

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Coursera</th>
<th>edX</th>
<th>Udacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launched</td>
<td>April 2012</td>
<td>April 2012</td>
<td>February 2012</td>
</tr>
<tr>
<td>Founding Partners</td>
<td>Andrew Ng and Daphne Koller, two CS professors from Stanford</td>
<td>MIT and Harvard</td>
<td>Sebastian Thrun, David Stavens and Mike Sokolsky, originally all from Stanford</td>
</tr>
<tr>
<td>Categories</td>
<td>20 CS related</td>
<td>unknown</td>
<td>3</td>
</tr>
<tr>
<td>Course</td>
<td>4 CS related</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>Time release/Self study</td>
<td>both</td>
<td>time</td>
<td>self</td>
</tr>
<tr>
<td>Certificate of Completion</td>
<td>Not for selfstudy</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Partners</td>
<td>33</td>
<td>6</td>
<td>No official university partners</td>
</tr>
</tbody>
</table>
For each course, the following information was captured from the course websites:

- a unique number was assigned, $Cn$ for Coursera, $En$ for Edx and $Un$ for Udacity
- the course title and/or code
- the affiliated institution
- the prerequisites
- the outcomes
For each course, the following information was determined:

- secondary school skill-set being addressed
- pedagogical setting rating
- whether the course provided a certificate of competence/attendance
- whether the course is self-study or presented in a specific time-frame
Introduction
Requirements for Computer Science
University entrance requirements for CS
Online courses
Course selection
Conclusion

Course providers
Classification of introductory CS courses

Secondary school skills distribution

CSERC 2013
Online courses for student success in CS
Pedagogical setting for a good online course:

- takes different learning styles into account (Visual, Auditory, Kinesthetic)
- encourages contact between instructor and students as well as between students
- facilitates active learning
- gives feedback and encourage according to expectations
- schedules activities
- fosters a strong sense of belonging online as online learning can be very lonely
<table>
<thead>
<tr>
<th>Equivalent courses</th>
<th>Skills</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 C2(5) and C10(4)</td>
<td>S1 - Algorithms</td>
<td>Foundation in programming required. C2 requires Java.</td>
</tr>
<tr>
<td>M3 C4(7), E1(4) and E2(3)</td>
<td>S2 - Analysis and Design</td>
<td>E2 includes basic programming, while C4 and E1 do not</td>
</tr>
<tr>
<td>M4 C6(4), C8(5), C9(3), E1(4) and U1(5)</td>
<td>S1 - Programming</td>
<td>Beginner programming, C8 and E1 are more Mathematical</td>
</tr>
<tr>
<td>M5 C7(5) and U5(5)</td>
<td>S2 - Testing</td>
<td>Both require a programming foundation</td>
</tr>
</tbody>
</table>
Introduction
Requirements for Computer Science
University entrance requirements for CS
Online courses
Course selection
Conclusion

Course equivalences
Proposed curriculum

CSERC 2013
Online courses for student success in CS
Introduction
Requirements for Computer Science
University entrance requirements for CS
Online courses
Course selection
Conclusion

Course equivalences
Proposed curriculum

CSERC 2013
Online courses for student success in CS
Online introductory courses focus mainly on programming and algorithms

- Limited coverage of secondary school skills still have a positive contribution to undergraduate skills
- Self-study vs teacher guided study, influenced by learner maturity
- Following a curriculum of online courses will have a positive influence on first year undergraduate throughput
- With no guarantee of course delivery, the proposed online course curriculum needs to be reviewed annually
- Universities need to develop their own admissions tests if they wish to use them and do their own analysis of the results
- Maturity in online courses required, particularly with regards to the pedagogical setting