Faculty Review of Open eTextbooks

The California Open Educational Resources Council has designed and implemented a faculty review process of the free and open etextbooks showcased within the California Open Online Library for Education (www.cool4ed.org). Faculty from the California Community Colleges, the California State University, and the University of California were invited to review the selected free and open etextboks using a rubric. Faculty received a stipend for their efforts and funding was provided by the State of California, the William and Flora Hewlett Foundation, and the Bill and Melinda Gates Foundation.

Textbook Name:

How to Think Like a Computer Scientist: Learning with Python 3

Find it: eTextbook Website

Textbook Authors:
Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyer

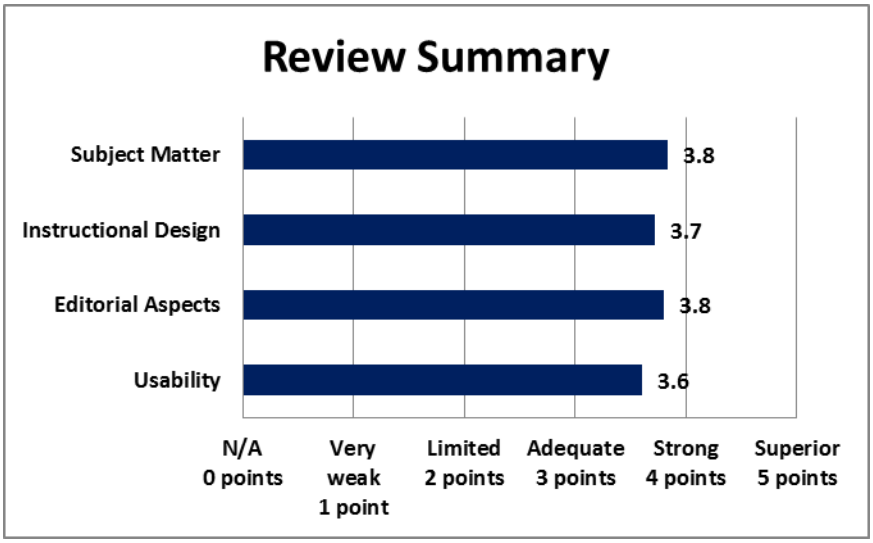
Reviewed by:
Garrett Whelan

Institution:
Long Beach City College

Title/Position:
Professor

Format Reviewed:
Online

A small fee may be associated with various formats.



Date Reviewed:
March 2015

California OER Council eTextbook Evaluation Rubric

CA Course ID: COMP 122

Subject Matter (30 possible points)	N/A (0 pts)	Very Weak (1pt)	Limited (2 pts)	Adequate (3pts)	Strong (4 pts)	Superior (5 pts)
Is the content accurate, error-free, and unbiased?					X	
Does the text adequately cover the designated course with a sufficient degree of depth and scope?					X	
Does the textbook use sufficient and relevant examples to present its subject matter?				X		

Does the textbook use a clear, consistent terminology to present its subject matter?					X	
Does the textbook reflect current knowledge of the subject matter?					X	
Does the textbook present its subject matter in a culturally sensitive manner? (e.g. Is the textbook free of offensive and insensitive examples? Does it include examples that are inclusive of a variety of races, ethnicities, and backgrounds?)					X	

Total Points: 23 out of 30

Please provide comments on any aspect of the subject matter of this textbook:

- It's a stripped down book, but Python is a relatively minimalist language so I think that works in this case. It feels like there isn't much content in this book, but it does cover everything that is required per the C-ID (except language comparison, but that is pretty minor).

- Everything is there, it's correct, and it's well explained.

Instructional Design (35 possible points)	N/A (0 pts)	Very Weak (1pt)	Limited (2 pts)	Adequate (3pts)	Strong (4 pts)	Superior (5 pts)
Does the textbook present its subject materials at appropriate reading levels for undergrad use?					X	
Does the textbook reflect a consideration of different learning styles? (e.g. visual, textual?)					X	
Does the textbook present explicit learning outcomes aligned with the course and curriculum?				X		
Is a coherent organization of the textbook evident to the reader/student?					X	
Does the textbook reflect best practices in the instruction of the designated course?					X	
Does the textbook contain sufficient effective ancillary materials? (e.g. test banks, individual and/or group activities or exercises, pedagogical apparatus, etc.)				X		
Is the textbook searchable?					X	

Total Points: 26 out of 35

Please provide comments on any aspect of the instructional design of this textbook:

- The learning outcomes are not "explicit," but I'm not bothered by that. Learning programming does not require that the students know the formal terminology for their outcomes, I would rather have them focus on being able to write the code.

- As for best practices and ancillary materials, please see the comment on what I would highlight to colleagues. There is another, interactive book built upon this text which is much better.

Editorial Aspects (25 possible points)	N/A (0 pts)	Very Weak (1pt)	Limited (2 pts)	Adequate (3pts)	Strong (4 pts)	Superior (5 pts)
Is the language of the textbook free of grammatical, spelling, usage, and typographical errors?					X	
Is the textbook written in a clear, engaging style?					X	
Does the textbook adhere to effective principles of design? (e.g. are pages latid0out and organized to be clear and visually engaging and effective? Are colors, font, and typography consistent and unified?)				X		
Does the textbook include conventional editorial features? (e.g. a table of contents, glossary, citations and further references)					X	
How effective are multimedia elements of the textbook? (e.g. graphics, animations, audio)					X	

Total Points: 19 out of 25

Please provide comments on any editorial aspect of this textbook.

- The book is very readable, which is hugely important for community college students. It was written by a computer scientist without an editor, so I have to say that the visual impact and design is both clunky and drab, but much of that is corrected in the interactive book (please see comments on what I would highlight to colleagues).

Usability (30 possible points)	N/A (0 pts)	Very Weak (1pt)	Limited (2 pts)	Adequate (3pts)	Strong (4 pts)	Superior (5 pts)
Is the textbook compatible with standard and commonly available hardware/software in college/university campus student computer labs?					X	
Is the textbook accessible in a variety of different electronic formats? (e.g. .txt, .pdf, .epub, etc.)					X	
Can the textbook be printed easily?					X	
Does the user interface implicitly inform the reader how to interact with and navigate the textbook?				X		
How easily can the textbook be annotated by students and instructors?				X		

Total Points: 18 out of 30

Please provide comments on any aspect of access concerning this textbook.

- It's a little clunky to get their tools working which may be a barrier to students without a great deal of technical proficiency, but a 3-4 page set of instructions should be adequate to fix that.
- There is no provision for annotating the HTML or PDF, but both can be downloaded if that is required.

Overall Ratings						
	Not at all (0 pts)	Very Weak (1 pt)	Limited (2 pts)	Adequate (3 pts)	Strong (4 pts)	Superior (5 pts)
What is your overall impression of the textbook?					X	
	Not at all (0 pts)	Strong reservations (1 pt)	Limited willingness (2 pts)	Willing (3 pts)	Strongly willing (4 pts)	Enthusiastically willing (5 pts)
How willing would you be to adopt this book?					X	

Total Points: 8 out of 10

Overall Comments

If you were to recommend this textbook to colleagues, what merits of the textbook would you highlight?

- This is not the book you should be using for your introductory programming class. You should be using the interactive textbook based on this book: http://interactivepython.org/courselib/static/thinkcspy/toc.html
- It's buried in the resources for this text, but it fixes many of the shortcomings of the book with:
  - Interactive and embedded programming exercises in the text
  - Mini-quizzes to test for comprehension, again embedded in the text
- By combining the interactive version online with the PDF version to read/reference, you have a very solid basis for a class.

What areas of this textbook require improvement in order for it to be used in your courses?

- It doesn't have slides or test banks, but that's minor. The largest structural defect is the lack of assignments that the students can self grade.
- It's best when I can assign students 25-50 small programming exercises per chapter that they can work on without me being present. That means that either the correctness of their output has to be obvious, or there has to be some online grading system. Currently the textbook that I'm using has a resource for that (http://practiceit.cs.washington.edu/index.jsp) and any open book would need the equivalent.
- The interactive book goes a long way toward closing that gap, but it's not quite there.

CALIFORNIA OER COUNCIL
CSU + CCC + UC

For questions or more information, contact the CA Open Educational Resources Council.