Faculty Review of Open eTextbooks

Textbook Name:

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

License:
How to Think Like a Computer Scientist: Learning with Python 3 by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers is Copyright, licensed under a GNU Free Documentation License.

Find it: eTextbook Website

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

Reviewed by:
Daniel Frost

Institution:
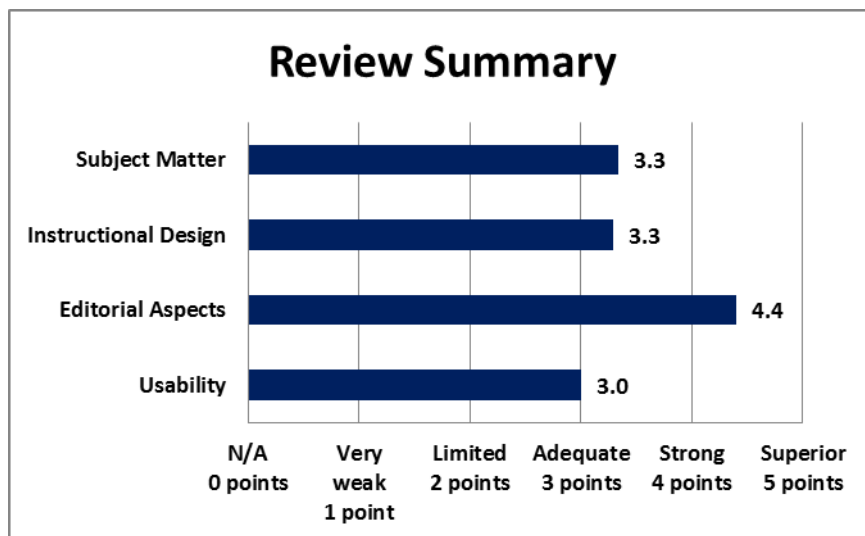University of
California, Irvine

Title/Position:
Professor

Format
Reviewed:
Online

A small fee may be associated with various formats.

Review Summary

Category	Score
Subject Matter	3.3
Instructional Design	3.3
Editorial Aspects	4.4
Usability	3.0

N/A 0 points | Very weak 1 point | Limited 2 points | Adequate 3 points | Strong 4 points | Superior 5 points

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	

Question	N/A (0 pts)	Very Weak (1pt)	Limited (2 pts)	Adequate (3pts)	Strong (4 pts)	Superior (5 pts)
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: 20 out of 30

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

- This book uses Python 3, and the link we were given points to the "Rhodes Local Edition." I'm not sure precisely what that refers to, but it does mean that the book is tailored to the needs of a specific college or university.
- Some choices that work for the author, such as using the PyScripter IDE and the Turtle module, are not widely adopted and probably limit the usefulness of this book.
- Otherwise, the book covers a good range of COMP 122 material, particularly category I, Programming Fundamentals.
- I like the Glossary at the end of each chapter.
- The Exercises are fine – more would be useful, but I imagine most instructors will want to create their own homework assignments anyway.

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: 23 out of 35

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

- The book is completely adequate as regards Instruction Design, but certainly does not go above and beyond the minimal requirements. For instance, the web version doesn't take advantage of the web's interactive capabilities.

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: 22 out of 25

Please provide comments on any editorial aspect of this textbook.

- Nicely laid out and visually interesting.

Usability (25 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: 15 out of 25

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

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: 7 out of 10

Overall Comments

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

- The book is a fine introduction to programming in Python 3. I'm not sure it really lives up to the "How to Think Like a Computer Scientist" title, but it does introduce the student to a good range of CS topics in the context of beginning Python programming, such as program design and data structures. The language is lively and the pages are attractive to look at.
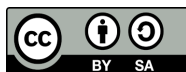
What areas of this textbook require improvement in order for it to be used in your courses?

- I'd like to see hands-on exercises at the end of every section (not every chapter). The book covers a lot of material, and doesn't sufficiently accommodate the language learner's need to practice, practice, practice.

We invite you to add your feedback on the textbook or the review to the textbook site in MERLOT
(Please register in MERLOT to post your feedback.)

CALIFORNIA
OER
COUNCIL
CSU+CCC+UC

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