

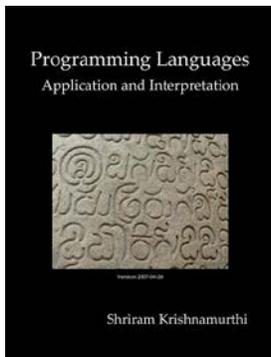


Faculty Review of Open eTextbooks

The [California Open Educational Resources Council](http://www.cool4ed.org) 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:

Programming Languages: Application and Interpretation



License:



Programming Languages: Application and Interpretation by Shriram Krishnamurthi is licensed under [Creative Commons Attribution-NonCommercial-ShareAlike 3.0](https://creativecommons.org/licenses/by-nc-sa/3.0/)

Find it: [eTextbook Website](#)

Textbook Authors:

Shriram Krishnamurthi

Reviewed by:

Fay Zhong

Institution:

California State University, East Bay

Title/Position:

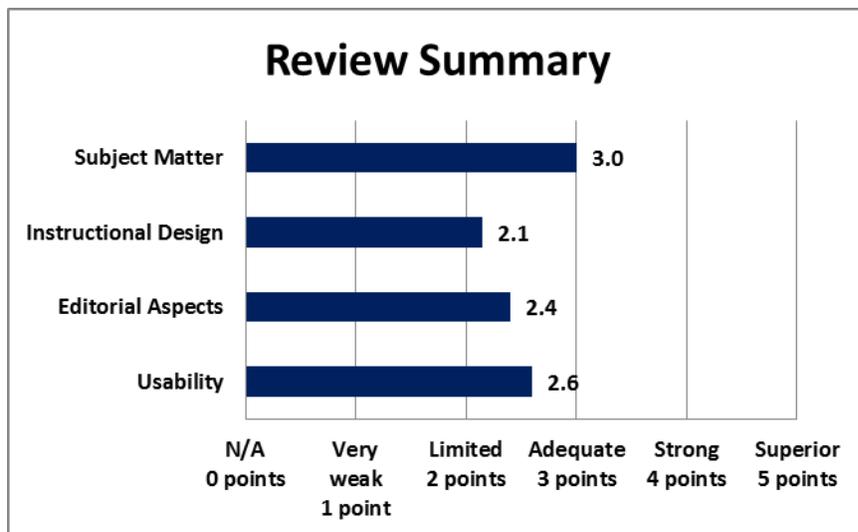
Professor

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: 18 out of 30

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

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	X		
Is a coherent organization of the textbook evident to the reader/student?						
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: 15 out of 35

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

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 laid out 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: 12 out of 25

Please provide comments on any editorial aspect of this textbook:

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: 13 out of 30

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: 3 out of 10

Overall Comments

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

- This is a hard-core book that covers piles of topics in programming concepts. The main programming language used in this book is Racket, which is a “full-spectrum programming language”, “goes beyond Lisp and Scheme with dialects that support objects, types, laziness, and more.”
- The book covers many important concepts in Programming Concepts & Methodology, such as parsing (Ch. 2), interpreter (Ch. 3, Ch. 5), data representations and functions (Ch. 5), variables and parameter passing (Ch. 8), recursion (Ch. 9), objects (Ch. 10), memory management (Ch. 11), types (Ch. 15), etc.
- This book does not follow a top-down narrative. It has the flow of a conversation, with backtracking: build up programs incrementally, present each “obvious” version and show why it fails, and use that failure to iterate. Include mistakes make it impossible for people to read passively. The author wants to show students where languages come from, why we should regard languages as the ultimate form of abstraction, how to recognize such an evolving abstraction, and how to turn what they recognize into a language.
- The book has plenty of examples and sample code, so students can understand the idea or concept with more concrete impression.

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

- As the author mentioned in the book, “Racket’s syntax, which it inherits from Scheme and Lisp, is controversial.”
- The book does assume that a student can write rudimentary programs in Scheme. However, it does not begin with a Scheme tutorial.
- The author left out object-oriented programming from this book.
- This book introduces many details of Racket; however, Racket is not a very popular language and thus might make the learning process a bit steep.
- The structure of the book, i.e. the non-linear path, is a little bit frustrating, which makes the book “a poor reference guide” (as mentioned by the author): “you can’t open up to a random page and be sure what it says is correct.”
- There are some minor typos, for example, in “11.5 Conservative Garbage Collection”: “Conservative” should be “conservative”. In “8.1.5 Can the Environment Help”: “both branches of the sequence”, one “both” should be deleted. In “8.1.8 The Bigger Picture”: “that will can” should be “that will”.
- More multimedia elements, e.g. pictures, animations, audio, video lectures are desired in the textbook.

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.)



For questions or more information, contact the [CA Open Educational Resources Council](#).



This [review](#) is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](#).