

**DESIGNING AN EFFECTIVE INTERACTIVE E-BOOK FOR
COMPUTER SCIENCE EDUCATION**

A Thesis
Presented to
The Academic Faculty

by

Steven James Moore

In Partial Fulfillment
of the Requirements for the Degree
Computer Science in the
School of Interactive Computing

Georgia Institute of Technology
December 2014

ACKNOWLEDGEMENTS

I wish to thank my Advisor, Dr. Mark Guzdial for his guidance and support. A special thanks to everyone in the Contextualized Support for Learning Lab for providing me with assistance and review. Thank you for all the help and support!

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABSTRACT	ix
<u>CHAPTER</u>	
1 Introduction	10
2 Literature Review	14
3 Methodology	18
Usability Methodology	18
Participants	18
Materials	18
Instrumentation	19
Procedure	20
Learnability Methodology	21
Participants	21
Materials	22
Instrumentation	22
Procedure	23
4 Results	25
Usability Results	25
Preferences for the Runestone Platform	25
Comparing E-book Platforms	27

Improving E-books	28
Learnability Results	30
Participants' Learning	30
Usefulness for Teachers and Students	31
5 Discussion	33
Runestone	33
Comparing Runestone, Zyante, and CS Circles	36
Improving Design and Usability	36
Learnability	39
Limitations	41
Conclusion	41
REFERENCES	43
APPENDIX A: Usability Survey	45
APPENDIX B: Learnability Pretest	72
APPENDIX C: Learnability Chapters 1 & 2 Pretest	76
APPENDIX D: Learnability Chapters 3 & 4 Pretest	78
APPENDIX E: Learnability Chapters 5 & 6 Pretest	80
APPENDIX F: Learnability Chapters 7 & 8 Pretest	84

LIST OF TABLES

	Page
Table 1: Numbers of Pretest and Posttest Questions	23
Table 2: Average Design Factor Ratings	27
Table 3: Widget Platform Preference	27

LIST OF FIGURES

	Page
Figure 1: Runestone Design Factor Ratings	25
Figure 2: Reported Widget Usefulness	30
Figure 3: Pretest and Posttest Scores	31

ABSTRACT

There's a need to develop more computer science teachers around the world. As massive open online courses have been failing, an interactive e-book used for distance learning might fulfil this need. This research seeks to determine what makes for good usability in a computer science e-book and measuring if teachers learn using them. The study investigates participants' preferences for interactive computer science e-book designs and usability, in an attempt to review and develop guidelines for educational e-book creation. Comparisons and preferences were made between three interactive educational computer science e-books and specific multimedia widgets within them. Based on the reported findings, some general guidelines were suggested for increasing usability within interactive e-books and ways to enhance their educational value. The results of the study suggest that interactive educational e-books are an effective alternative to current distance learning methods for teaching computer science.

CHAPTER 1

INTRODUCTION

The shift of technology in the 21st century has led to education incorporating more technology-based curriculums. By 2015, the U.S. aims to have 10,000 teachers in 10,000 high schools teaching a new computer science curriculum as part of the CS10k initiative (Astrachan et al., 2011). One proposed method to meet this goal is by using an e-book to help teachers learn the subject matter (Guzdial, 2013). An e-book can fit into the busy lifestyle of such individuals, as they can learn at their own pace and schedule. To facilitate and assist in this learning process, the educational e-book will be interactive. It will utilize various multimedia technologies, such as videos, audio, and configurable widgets. The usability of these components is essential in ensuring an interactive e-book is both user friendly and an effective educational resource.

For an interactive educational e-book to serve as an effective digital teaching platform, the users must be able to efficiently learn from using the e-book. The usage of the e-book should contribute to the user's content knowledge on the subject. Content knowledge generally refers to the facts, concepts, theories, and principles that are taught and learned. Traditional textbooks are limited to immobile images and text on the pages to communicate the knowledge they contain. However, interactive educational e-books have the capability of using multimedia technologies to enhance the teaching of content knowledge. The design and usability of the multimedia technologies within these interactive educational e-books largely influences their ability to serve as an effective teaching platform.

Prior research on the usability of e-books is generally limited to non-interactive ones and focuses more on the overall design of the e-book, rather than focusing on specific components of it. A study by Landoni et al. (2000) developed a series of design tactics to follow in order to produce an effective e-book. Many of these design tactics, such as proper titles, pagination, and typographical aspects, are still used in modern e-book production. Another similar study investigated and surveyed undergraduate students' design preferences for three different e-books (Chong, Lim, & Ling, 2009). This study also resulted in a combination of suggested design heuristics to follow when designing an e-book.

Previous studies on content knowledge with e-books is both limited and divided. A study conducted by Campbell et al. (2014) found that the use of e-books could improve learned content knowledge due to the active learning and portability the e-books provide. In contrast to this study, a study on web-based learning that analyzed educational websites and resources determined that these tools were an advancement for technology, but a setback for pedagogy (Mioduser et al., 2000). They found that the educational websites took advantage of technology's abilities to enhance communication and provide better information representation. However, the sites were not developed with beneficial pedagogical approaches, such as active involvement and scaffolding, as a basis. While these two studies have contrasting results, it is clear that e-books have a crucial impact on content knowledge and how it is learned. One study, which merged the concepts of e-book usability and content knowledge, intended to determine the characteristics of e-book creation that afford learning and educating (Wasecka, 2013). The results found that a combination of several characteristics, such as feedback, creativity, and productivity, could ultimately be used in the creation of an

e-book that properly affords learning.

While there exists research regarding the design of standard e-books, there is a limited amount of research discussing the design and usability of interactive e-books. This limitation also exists for research about learning and measuring content knowledge from interactive educational e-books. Studies exist analyzing the educational value of e-books and comparing them to printed books, but few studies researched interactive e-books for their educational merit. Research that does analyze interactive e-book design and usability rarely does so across multiple platforms or similar e-books, but rather uses a single interactive e-book for the study.

There is a current unmet educational need for individuals, such as teachers in the CS10k initiative, to learn computer science. Providing an effective means to learn computer science via an interactive educational e-book can help meet this need. This makes research connecting the usability and educational effectiveness of such e-books beneficial and relevant.

The current study's intent is to analyze the usability of interactive components within interactive educational e-books and determine how usability is correlated with the content knowledge gained from using the e-book as a teaching platform. This research was conducted in two separate studies at the Georgia Institute of Technology that consist of administering interactive surveys regarding e-book usability and content knowledge. By determining user friendly interactive components that are also efficient in teaching, this study can contribute to the knowledge on effective digital teaching platforms and can help meet the need for computer science instruction. In addition, contributions can be made to the modern design suggestions for interactive components of e-books that are currently present. Also, we wanted to determine how

user friendly and educationally effective the specific multimedia tools used in our e-book were compared to that of other similar e-books. This project enables us to provide insight and improvement regarding usability guidelines for multimedia tools within interactive e-books.

CHAPTER 2

LITERATURE REVIEW

Previous research regarding the design preferences for e-books was mostly concentrated on non-interactive e-books. Much of the research in this area intends to establish a set of guidelines and suggestions for designing an e-book. As a result there are several common themes for e-book design that are shared between many of the studies. For instance, maintaining a resemblance to a traditional printed book is noted as being an important design feature. Displaying the e-book's content on pages in a model that resembles physical books assists the user in feeling comfortable and experienced with the e-book (Wilson, Landoni, & Gibb, 2000). Wilson, Landoni, & Gibb found that this association enforced a recognizable logical structure to the book that allowed the reader to better gather information. A similar study was conducted by Berg et al. (2010) that investigated how students used e-books compared to print texts. The study identified that the information retrieval behavior of students was enhanced by the e-book resembling a printed book. This study also concluded that navigation plays a large role in information retrieval and enhances the usefulness of an e-book as a whole.

Further studies expand on these design guidelines by providing graphical user interface specifics, such as page layout, font weight, and white space to use in the construction of an e-book (Chong, Lim, & Ling, 2009). Adhering to these specifics creates a higher ease of use, which was found to be associated with a higher ease of both navigation and scanning of the e-book's text. Researchers concluded that following these specifics, such as having a consistent font size, can be used to improve

the navigation design, page layout, and content design of e-books. To expand on the navigation design, related research by Chowdhury (2004) found that the inclusion of a search option greatly improved a user's ability to navigate and retrieve information from an e-book. The lack of this search option was reported to be a common problem that negatively influenced e-book usability and overall reception. They concluded that if a search tool or widget is provided, then it should be similar to the ones found in common web technologies that the user is likely to be familiar with.

Previous studies focus on the design of interactive e-books and learning experience gained by the user through interactions. The majority of these studies offer reviews on the general effectiveness of specific interactive e-book elements as reported by participants. These studies were similar to the ones about e-book design, but focused specifically on the design of interactive components within the e-book rather than the e-book as a whole. One such study focused on the engagement provided by the multimedia tools in an interactive e-book, by using animations to grab and maintain attention (Hamed & Hosam, 2013). This study reviewed multiple multimedia tools in e-book design and found that providing interactivity was beneficial to student learning. Another study found that the use of an interactive e-book over a standard printed textbook increased learnability, through the form of increased test scores, in the majority of the study's student participants (Torigoe, 2013).

A study similar to this current one was conducted by Fenwick et al. (2013), which designed an interactive e-book for computer science instruction and surveyed students' responses to the interactive components of the e-book. It detailed the interactive aspects used in the e-book, such as embedded slideshows, and reported the participants' attitudes toward them. They found that videos were the most favored interactive component of

their e-book and that coding widgets were found to be difficult and hard to type code on. Another study found that test scores increased by seventy-three percent for the participating students who used an interactive e-book over a printed text book in an introductory college computer science course (Torigoe, 2013). These studies provide support for the use of interactive e-books over both non-interactive e-books and printed texts, with certain preferences toward specific e-book interactivity.

According to Shulman (1987), pedagogical content knowledge (PCK) is “the blending of content and pedagogy into an understanding of how particular topics, problems or issues are organized, represented, and adapted to the diverse interests and abilities of learners, and presented for instruction” (p. 4). While research on this type of content knowledge is abundant, its application to the use of e-books in particular is quite miniscule. A study by Niess (2005) found that the integration of technology with teaching, such as through the use of e-books, was beneficial in advancing PCK in science, technology, engineering, and mathematics (STEM) fields of study. From these results they inferred that the use of technology can increase the learnability of subject matter.

Studying the usability of interactive e-books as well as the content knowledge gained from them is important because this research has implications for the future of computer science education. Previous research has already established a solid baseline for designing a non-interactive e-book. However, as technology improves and e-books continue to grow in popularity, interactive e-books will become more prevalent. There is an obvious lack of research in interactive e-books, in both their usability specifics and educational merit. The few existing studies about interactive e-books conclude that the interactive components within the book are beneficial and can increase learnability

among other benefits. These studies rarely compare interactive e-books across multiple platforms or ones that cover a similar topic, such as computer science. Properly taking advantage of an e-book's interactivity to improve content knowledge has influential educational value not just for computer science, but across multiple disciplines.

CHAPTER 3

METHODOLOGY

Usability Study

3.1 Participants

Subjects consisted of 15 male and 7 female teachers that had six months or more of prior coding experience and knowledge. Additionally, all participants held at least a bachelor's degree and had previously used an e-book.

3.2 Materials

Three interactive educational computer science e-books that are accessible through a user's web browser were used for this study. All three e-books are designed to teach the Python programming language and concepts at an introductory level. While the overarching concept of the three e-books is similar, their design and layout are different. To conceal the e-books' identities they will be addressed by the platforms on which they were created on. The first e-book will be referred to as Runestone, the second Zyante, and the third CS Circles.

In addition to the overall design of three e-books, four interactive components found in each e-book were studied. These interactive components are known as widgets and allow the user to interact in with them in some form. The first widget is known as Active Code, which allows the user to edit and execute Python code, displaying any results or output, in the web browser. The next widget acts as a code visualization tool that allows the user to step through the code. It also displays variable values and program output, it's known as Code Lens. Parsons Problems is the third widget, which

allows the user to drag and drop blocks of code, from a bank of code blocks, into the correct order. Finally, the fourth widget is referred to as Multiple Choice which asks the user questions about a coding concept or output and can be answered as A-D or true and false. It provides varying levels of feedback for why the selected answer is correct or incorrect.

A survey was used to obtain the participants' feedback on the design and usability of the e-books and their interactive widgets. A copy of the survey and all its questions can be viewed in Appendix A.

3.3 Instrumentation

An online questionnaire approach was employed to study the participants' preferences on the varying usability and the three e-books and their interactive widgets. The four-part questionnaire began with asking basic demographic information of the participants. In this section, participants also reported their prior e-book and coding experience.

The second part gauged the participants' overall design preferences for the three e-books as a whole using a five point Likert scale. They were to rate the three e-book designs on the following factors: navigation, page information, media arrangement, page layout, font, legibility, white space, and color contrast between background and content. Additionally, they were asked to provide any feedback they had toward the design choices of the e-book via a free response question. For this section, the participants were provided with a URL and asked to rate the Runestone e-book first, then the Zyante e-book, and lastly the CS Circles e-book.

The third part determined the participants' usability and learnability preferences for the four interactive widgets found in each of the e-books. Participants were solicited

for feedback regarding the four widgets and their corresponding e-book platforms. For this section the Active Code widget was inquired about first, followed by Code Lens, then Parsons Problems, and finally Multiple Choice. The corresponding e-book platforms went in the same order for each of the four widgets, with Runestone being first, followed by Zyante, and then CS Circles. For example, Active Code implemented on Runestone was inquired about first, then Active Code on Zyante, and finally Active Code on CS Circles.

For each widget-platform combination, a URL to a web page containing that specific widget implemented on the specific platform, Runestone, Zyante, or CS Circles, was provided to the participant. After interacting with the widget, they were asked to state the purpose of the widget. They were then asked to describe what they think each button and feature of the widget does. Finally, were always asked to report anything they found confusing or didn't particularly like about the widget.

The final and fourth part of the questionnaire asked the participants to report which platform they thought implemented which widget the best. They compared the widgets on different platforms to one another, such that Code Lens on Runestone, Zyante, and CS Circles was compared against one another. After selecting their favorite widget-platform combination, the participants were asked to explain why they made their selection. The online questionnaire used for this survey was hosted via the online survey platform known as SurveyMonkey. A combination of preformulated response item set and open-ended format set was used throughout the questionnaire. Again, a copy of the survey and all its questions can be viewed in Appendix A.

3.4 Procedure

The participants were invited to take part in the study via an initial recruitment

email. This email included the purpose of the research and the specific criteria of participants that we were seeking. The criteria for this study was that participants must have six months or more of programming experience with a textual language. Upon receiving an email response from the potential participants stating that they would like to participate, participants were provided with a link to the survey via another email.

The survey contained clear written instructions prompting the user to complete the first part of the questionnaire gathering background information. Next, participants were provided links to the three different e-books and told to view and interact with them to their preference. Following this, the participants were to complete the second part of the survey where they answered questions regarding their design preferences of the three e-books as a whole. They were then provided links to the individual interactive widgets contained in each of the e-books. Once again, the participants were instructed to interact with the widgets to their desire. Participants were then asked a series of usability questions corresponding to the particular widget they had just interacted with. For the final part of the survey, they were asked to compare the widgets to one another, report which was the most useful to them, and explain why.

The data collected was processed and analyzed using Microsoft Excel 2013. Frequency distributions were computed for each design item and then compared across all three e-books. Participants' free responses and suggestions regarding the e-books were coded and analyzed manually.

Learnability Study

3.5 Participants

Subjects consisted of 2 males and 1 female, all of whom were teachers that had two months or fewer in prior coding experience and knowledge.

3.6 Materials

An interactive educational computer science e-book that's accessible through a user's web browser was used for this study. It was developed using the Runestone Interactive platform and is intended to teach the Python programming language and concepts at an introductory level to teachers. Only chapters one through eight of the e-book were used for this study. These chapters cover introductory computing concepts in Python, such as naming variables and repeating code with while and for loops. A pretest was used to gauge the participants' prior programming knowledge. Four posttests were also used to test how much the participants learned from using the e-book. The pretest and four posttests can be viewed in Appendixes B through F respectively.

3.7 Instrumentation

An online questionnaire approach consisting of five different surveys was employed to study the users' acquisition of knowledge after using the e-book. The first of the online questionnaires was a two part pretest, with the first part soliciting the participants for basic demographic information and to report their prior experience with any programming languages. The second part consisted of nine free response questions based on five programming problems. These questions were intended to further gauge the participants' prior programming knowledge and establish a baseline of programming knowledge. For instance, a block of code would be presented to the participant and they'd answer what values would be printed out when that block of code was executed.

The other four surveys are two part posttests designed according to the content covered in the e-book. Each covers two consecutive chapters of the e-book, beginning with chapter one and ending with chapter eight for a total of four posttests. These posttests ask a series of three to four multiple choice questions about a block of code or

specific concept learned in the previously read two chapters of the e-book. The questions were designed to assess the participants' learned content knowledge from using the e-book.

Following these three to four questions, the second part of each posttest consisted of three questions about the participant's favorite feature, what'd they change in the e-book, and if they felt like the e-book would be a beneficial tool for students and teachers. The final posttest for chapters seven and eight contained two additional questions in this section. They asked the participant to rank their confidence level in teaching the material on a five point Likert scale and if they could be provided with anything else that'd benefit their learning process. Table 1 shows a breakdown of the asked questions regarding the learned book material and participant preferences for the pretest and each of the posttests. All the posttests and the pretest were hosted via the online survey platform known as SurveyMonkey. Again, see Appendixes B through F for a copy of the survey and questions used for this study.

	Pretest	Posttests			
	Pretest	Chapters 1 & 2	Chapters 3 & 4	Chapters 5 & 6	Chapters 7 & 8
Book Questions	9	3	3	4	3
Preference Questions	0	3	3	3	5

Table 1. The number of each type of question asked for the pretest and four posttests

3.8 Procedure

The participants were invited to take part in the study via an initial recruitment email. This email included the purpose of the research and the specific criteria of participants that we were seeking. The criteria for this study was that participants must have had two months or fewer in prior coding experience and knowledge. Upon receiving an email response from the potential participants stating that they'd like to participate, participants were provided with a link to the pretest via another email.

If the participant successfully filled out the pretest and met the participant criteria, they were then emailed a web link to the e-book and four posttests. They were instructed to read and interact with the first eight chapters of the e-book at their own pace. Upon reading two chapters of the e-book, they were then to complete the corresponding posttest. While completing the posttest, the participant was instructed to not refer back to the e-book or use any outside material that may assist them in answering the questions.

CHAPTER 4

RESULTS

Usability Study

4.1 Preferences for the Runestone Platform

Participants were asked to select which platform, Runestone, Zyante, or CS Circles, had their favorite implementation for each of the four interactive widgets. For this selection eighteen total responses were given. Runestone was chosen ten times for Active Code, seven for Code Lens, twelve for Parsons Problems, and ten for Multiple Choice. On average, participants ranked all the Runestone design factors as slightly above average, with all factors averaging between three and four on the five point scale. Figure 1 depicts the average design factor ratings for our e-book constructed on the Runestone platform. Navigation about the e-book was on average ranked the lowest ($M = 3.09$) and the color contrast between background and content ($M = 3.95$) was ranked the highest.

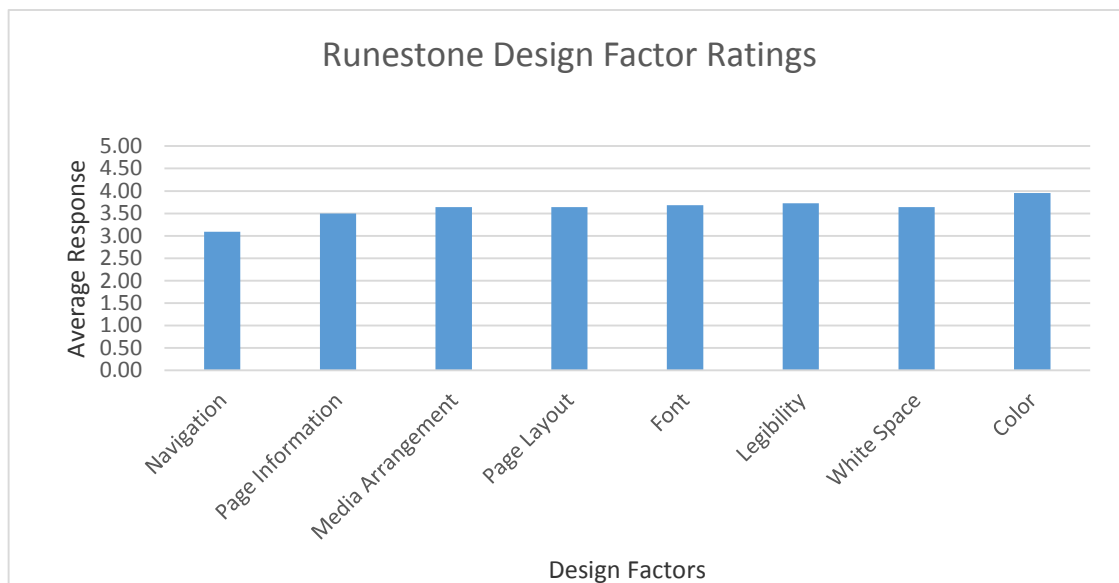


Figure 1. Average responses for the design factor ratings of the Runestone e-book

Participants were presented with four individual webpages each containing one of the widgets. They were asked to state what they believed the purpose of each widget was after interacting with it. For the purpose responses: eighteen of nineteen were correct for Active Code, sixteen of eighteen for Code Lens, fifteen of seventeen for Parsons Problems, and eighteen of eighteen for Multiple Choice. In total, only three of the participants misidentified the purpose of a widget. Of those three, one participant's purpose response was deemed indeterminate for three of the four widgets.

When participants chose their favorite platform implementation of specific widgets, they were encouraged to provide feedback on why they made their choice which was then coded around common themes. The most prevalent themes for the feedback of the participants that chose Runestone was that the Runestone design was both simpler and cleaner than the Zyante and CS Circles alternatives. Interesting insights came from the feedback provided regarding the favorite platform for implementing the Active Code and Multiple Choice widgets, respectively stated below:

“Cleanest looking, most ‘Python-like’ – simplicity and all that. The others look PC-ish (I’m a Mac user).”

“Looks the most modern and friendly, something that I’d want to use. Designs 2 [Zyante] and 3 [CS Circles] look like something out of a boring textbook.”

Similarly to the provided feedback on favorite widget-platform combinations, participants were instructed to provide comments on what they found confusing or didn't like for each of the four widgets presented on each of the three platforms. The Code Lens and Multiple Choice widgets implemented in Runestone received nearly no negative feedback in this section. However, the Runestone implementation of the Code Lens widget received the least amount of favorite votes for widget-platform

implementation. One participant provided a particularly insightful comment regarding why they did not like the Runestone Active Code widget:

“I wasn’t sure I could play with the code. I didn’t try.”

Another participant provided the following feedback for why they did not like the Runestone Parsons Problems widget:

“I’d have liked to see the code actually run (regardless of the order) so long as it was valid.”

4.2 Comparing E-book Platforms

Participants’ ranked design factors based on the overall design for each of the three e-books. The Runestone and Zyante platforms are similar in design rankings, while CS Circles is notably lower on most factors. Average design factor rankings for each platform are shown in Table 2 below.

	Runestone		Zyante		CS Circles	
	M	SD	M	SD	M	SD
Navigation	3.09	1.19	3.68	1.29	2.5	1.41
Page Information	3.5	0.8	3.43	1.12	2.73	1.12
Media Arrangement	3.64	1	3.5	0.96	2.86	1.08
Page Layout	3.64	0.95	3.67	0.97	2.73	1.12
Font	3.68	0.84	3.68	0.89	3.59	0.91
Legibility	3.73	0.7	3.68	0.78	3.5	1.19
White Space	3.64	0.9	3.5	1.19	3.27	1.2
Color	3.95	0.65	3.73	0.88	3.38	1.07

Table 2. Design factor averages and standard deviations on each e-book platform

In addition to ranking the overall design of the e-books, participants’ reported their favorite platform implementation of each widget, which is displayed in Table 3.

	Widget 1 - Active Code		Widget 2 - Code Lens		Widget 3 - Parsons Problems		Widget 4 - Multiple Choice	
	f	%	f	%	f	%	f	%
Runestone	10	55.56	7	38.89	12	66.67	10	55.56
Zyante	6	33.33	8	44.44	5	27.78	6	33.33
CS Circles	2	11.11	3	16.67	0	0	1	5.56
Total	18	100	18	100	18	94.45	18	94.45

Table 3. Platform preference for each of the four widgets, f denotes the number of participants

A single participant refused to pick a favorite platform for the Parsons Problems and Multiple Choice widgets, stating that they felt these widgets should not be used regardless of platform. Their feedback is provided below for Parsons Problems and Multiple Choice respectively:

“None of the above. They were all abysmal interfaces that would cause me to reject the book violently.”

“None of the above. I hate multiple choice and would not want to use any of these interfaces.”

Participants also provided feedback on why the widget-platform combination was their favorite. Feedback for why a specific implementation of Active Code was chosen as the favorite often involved the theme of the widget being clear and simple. For Code Lens, Parsons Problems, and Multiple Choice the most common response theme was based around the widgets’ provided feedback. Of the six times in total CS Circles was chosen for the favorite platform of a widget, it was due to the clear and simple design it had. While the participants’ reported their programming and interface design skill, it had no correlation with their widget-platform preference. Nor was there any significance between programming experience and interface design skill and the overall design rankings.

4.3 Improving E-books

Participants were asked to provide suggestions and ways to improve each of the four interactive widgets. Suggestions for the Active Code widget were primarily themed around providing more feedback, such as a detailed error message when the code fails to run. The majority of the suggestions for the Code Lens widget weren’t suggested improvement, but rather statements of how useful the widget is currently. For Parsons

Problems, several participants suggested showing the code's output, assuming it can be compiled and ran, regardless of it being in the correct order or not. Finally, suggestions for Multiple Choice revolved around them being used sparingly and for testing specific concepts.

The two worst ranked design factors for the Runestone platform were navigation and page information. Zyante's two worst design factors were page information and media arrangement. The two worst design factors for CS Circles were navigation and page information. An interesting quote regarding the navigation of Runestone from one participant is stated below:

"It's nice to see where you are in the course of a learning experience, and where you're going."

One participant provided an interesting quote regarding the amount of information on the page for the CS Circles platform:

"It was not very obvious what I had to do. There was more information below the task which needed to be read before the task."

In the top three highest ranked design factors for each platform, color contrast between background and content and font were present. Despite color being one of the highest ranked design factors across each platform, reports of the colors being too bright were present for each platform. This is also true for font, despite being reported as a favorite design factor, many individuals reported the font was too small in certain areas. Additionally, participants were asked to report how useful they believed each widget to be on a five point Likert scale. For the average reported widget usefulness, Active Code ($M = 4.56$) and Code Lens ($M = 4.5$) scored within .06 of one another and were ranked respectively as the two most useful widgets. Figure 2 displays the participants' average

widget usefulness ratings. There was no significant correlation with programming experience or gender and the perceived usefulness of the four widgets.

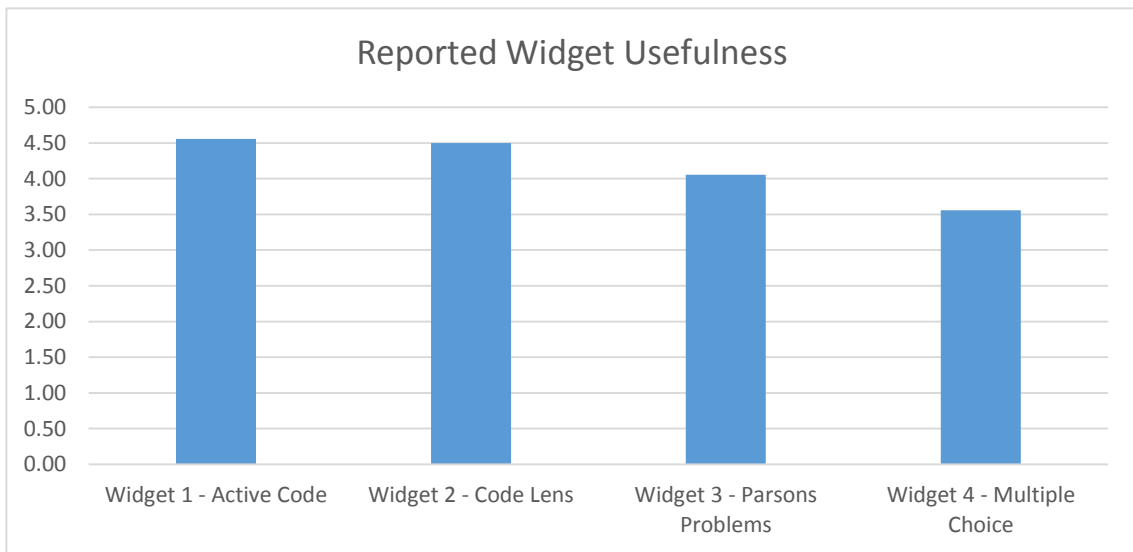


Figure 2. The average reported usefulness rankings for each widget

Learnability Study

4.4 Participants' Learning

Prior to interacting with the e-book, the three participants in the learnability study completed a pretest consisting of nine questions. Participants two and three correctly answered three of the nine questions and participant one correctly answered one of the nine questions.

Participant three only completed the pretest and first posttest, chapters 1 & 2, while participants one and two completed the pretest and all four of the posttests. Each of the posttests consisted of three to four questions, combined for a total of thirteen posttest questions. While participant one only correctly answered one of the nine pretest questions and participant two correctly answered three of the nine questions, they were respectively able to correctly answer a total of ten of thirteen and twelve of thirteen total posttest questions. Participant one increased from 11.11% on the pretest to 76.92% on the posttests and participant two increased from 33.00% on the pretest to 92.31% on the

posttests. Participant three correctly answered three of the nine pretest questions, increasing from 33.00% on the pretest to 100% on the chapters 1 & 2 posttest. The participants' percent correct on the five tests is shown in Figure 3.

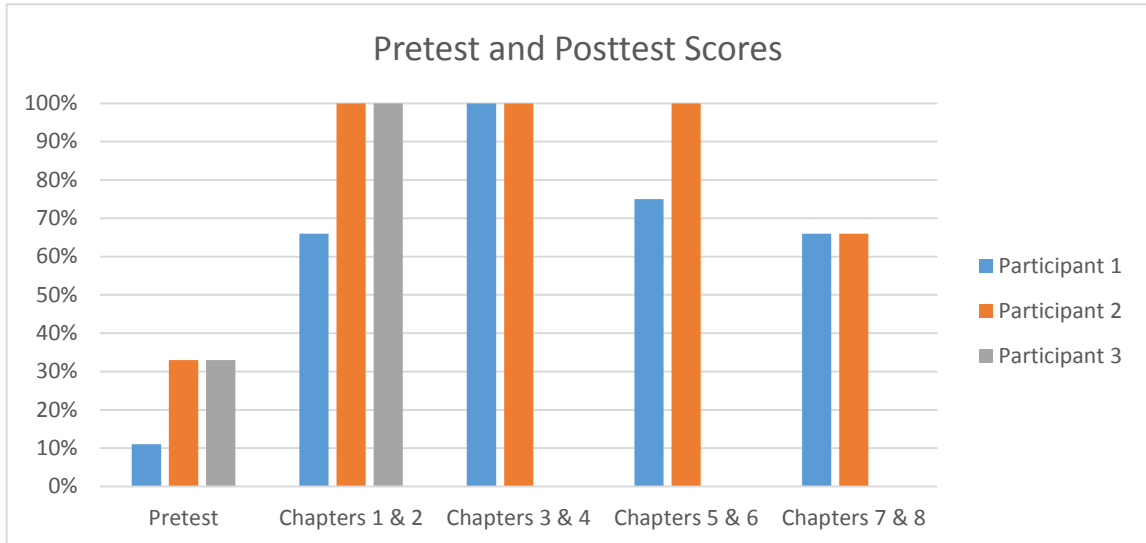


Figure 3. Participants' scores on the pretest and four posttests as percentages

Upon completing the last posttest, chapters 7 & 8, participants one and two were asked to rate their confidence in teaching the material that they've learned from the eight chapters on a five point Likert scale. Participant one reported having the highest (Excellent) confidence level in teaching the material and participant two reported having the middle (Average) confidence level.

4.5 Usefulness for Teachers and Students

At the end of each posttest, the participants were asked if they felt the e-book would be a beneficial tool for students and teachers. Of the nine instances the question was asked, every participant reported that they indeed thought the e-book would be an effective tool for both students and teachers. In particular, the participants stated this was in part to the interactivity and enjoyment from modifying the code featured within the e-book.

Participants were asked to report the most helpful feature of the e-book, to which

the ability to manipulate code and the provided interactivity were the most common responses. When asked if anything else could be provided that would benefit the learning process, participants one and two respectively responded:

“Live Chat during certain hours”

“Of course, I would benefit from a LOT more practice exercises to accompany the text. I feel like I get it, but won't hold on to it otherwise.”

CHAPTER 5

DISCUSSION

5.1 Runestone

For three of the widgets, Active Code, Parsons Problems, and Multiple Choice, Runestone was the most selected platform for their implementation. This suggests that participants found the Runestone implementation of those three widgets user friendly, or at least preferred them to two other popular platform implementations. The provided feedback from these participants for why the Runestone implementation was their favorite also indicates the widgets' current designs are understandable and clean. However, the Zyante implementation of the Code Lens widget was selected as the favorite widget-platform implementation by a single response over the Runestone version. The design of the Zyante Code Lens was quite different from the Runestone and CS Circles versions, as it only allowed the user to run the visualization and not step through each part of the code. Participants might prefer this limitation of the Zyante Code Lens, as their most common reason for selecting it as their favorite was due to it being both simple and clear. Many participants also stated that the Zyante Code Lens was the most visual walkthrough of the code.

On average, Runestone scored above three on a five point scale for all the design factors, with navigation being the lowest ranked design factor. While navigation's average score was still above average on the scale, it appears to be the design area of the Runestone e-book that needs the most improvement. Improving upon this design factor can create better usability with the e-book, as Berg et al. (2010) claims that navigation

plays an important role in the information retrieval process and can greatly enhance the usability of e-books. While navigation was on average ranked the lowest, color contrast between background and content was the highest ranked design factor for the Runestone platform. Interestingly, color was highly ranked for all three e-book platforms despite the different color palettes they each use.

The purpose for each widget implemented in Runestone was correctly perceived by the vast majority of the participants. Correctly identifying the purpose of the widgets is suggestive of their design being clear and straightforward, as also supported by much of the participant feedback. Participants who incorrectly stated the widgets' purpose did not necessarily misinterpret the purpose, but their answer was deemed indeterminate. It was sometimes the case that a participant stated what the code within the widget did when executed, rather than the intended purpose of the widget itself. It is quite possible the participants knew the widget's purpose but misinterpreted the question. This could cause the results to have a lower representation of the number of people who correctly know the purpose of a given widget, meaning certain widgets could be more useable than is currently represented by the results.

According to a majority of the participants' feedback, the clean and simple design of Runestone lead to its implementation of the widgets being chosen as the favorite widget-platform version. One participant stated the Runestone design appeared to be similar to the design of Apple's MAC over a Window's computer. This insight poses design alternatives, such as theming the e-book to match the operating system and device preexisting design heuristics, to enhance the e-book's usability. Tailoring the e-book's design in such a form could better blend it with the user's current experience, thus

creating a sense of coherence and enhancing overall usability. Another participant stated how the Zyante and CS Circles designs looked like something from a textbook, yet the design of Runestone did not. While Wilson, Landoni, and Gibb (2000) support adhering to the book metaphor, it appears that resembling a printed book can have adverse effects. Interestingly, no other participants made any comments or comparison of e-books to physical books in their provided feedback.

The lack of negative design and usability suggestions for the Multiple Choice widget within Runestone is not surprising, as it is similar to ones commonly found across other web technologies. However, the dynamic and tailored feedback for all three of the e-books' Multiple Choice widgets differentiates them from these commonly found web ones and increases their utility during the learning process. Despite Zyante's Code Lens being selected the most for favorite widget-platform combination, Runestone's implementation of Code Lens received minimal negative feedback which is indicative of it having at least average usability. It is interesting because the Zyante implementation of Code Lens is quite different from Runestone's implementation, as the Zyante version provides fewer options for interactivity. However, the similarity in the participants' feedback for the Runestone and Zyante Code Lens combined with their similar favorite widget-platform implementation results suggests that both designs are favorable to some degree. This lack of interactivity in the Zyante widget potentially makes the Runestone version the best choice. The Runestone implementation of Code Lens allows for increased interactivity and according to Hamed and Hosamn (2013) the interactivity of multimedia components benefits the learning process.

5.2 Comparing Runestone, Zyante, and CS Circles

While overall the design factor rankings for Runestone and Zyante were similar, a majority of the CS Circles design factors were on average ranked quite lower by comparison. The lower overall design factor rankings for the CS Circles design, compared to the Runestone and Zyante platforms, could in part be attributed to the extra features of the CS Circles widgets. Many of the CS Circles widgets had excess capabilities, such as buttons that only have a function when the code requires specific user input. Some participants were confused by the purpose of these features and thus ranked certain design factors for the CS Circles e-book lower, as stated in their feedback. This, along with participants' common theme for selecting Runestone widgets as their favorite, supports the need for the interactive components to be both clear and simple.

Many participants chose their favorite platform implementation of the Active Code widget due to it being clearly and cleanly designed. This feedback is similar to the feedback received during the e-book study by Chong, Lim, and Ling (2009), which found many participants favored e-book designs that did not complicate the page and were simplistic in design. Reasons for selecting the favorite platform implementation of the Code Lens, Parsons Problems, and Multiple Choice widgets were often themed around the feedback the widgets provided for correct and incorrect answers. This provided feedback of the widgets increases the usability and affordances to learning that the e-book offers (Wasecka, 2013).

5.3 Improving Design and Usability

Unlike the evaluations for the other three widgets, participant feedback for Active Code never involved it providing beneficial feedback for errors the user encountered

when editing the code. In fact, many participants stated the design of the Active Code widget is presently straightforward and maintains a clear appearance, yet lacks helpful feedback when they encountered an error. Similar to Wasecka's (2013) study, much of the provided participant responses indicated a need to improve feedback.

Code Lens lacked many suggestions that requested its design or functionality be changed, but mostly had participant feedback regarding how useful and well-designed it currently is. Design suggestions regarding this widget in particular should be taken from Runestone or Zyante, depending on what Code Lens style is appropriate for the e-book. A mixture of the two types of Code Lens widgets could offer an educational benefit that affords learning while maintain a high level of usability. Participant feedback for Parsons Problems also suggested increased feedback for correct and incorrect code block ordering, such as showing the code's output if it was able to be run. The common theme for providing more feedback for these widgets potentially suggests the widgets didn't always function as the participants thought, indicating a lower level of usability.

The last widget, Multiple Choice, was mostly deemed unfavorable or suggested that it should only be used for specific type of questions. This unfavorable attitude toward Multiple Choice may be attributed to the widget not being as innovative or new as the other three widgets. The Multiple Choice widget is comparable to multiple choice questions seen in printed textbooks, but offers dynamically provided feedback that printed works cannot. Interactivity within the Multiple Choice widget, regardless of platform implementation, is much more limited in comparison to the three other widgets. This limited interactivity and normalcy makes it not surprising that participants were generally not in favor of the Multiple Choice widget, despite their lack of negative

usability feedback for the widget or a distaste for Multiple Choice over coding for learning computer science. The dislike for the widget is not due to participants distaste for the widget's design, just for the concept of the widget as a whole.

Dissimilar to the study by Fedwick et al. (2013), no participants reported that the coding widgets were difficult or hard to type on. This implies they're sufficiently designed with adequate space for the user to type and interact with. While the design of the coding widgets did not receive any feedback regarding spacing, one design factor that did was the amount of information on the page. The amount of information on the page and navigation were on average ranked as the two lowest design factors across all three e-book platforms, suggesting they need improvement. Individual participant suggestions give insights on ways to help improve this, such as constantly displaying a table of contents to show the reader their current position in the e-book. The amount of information on the page being ranked so poorly is seemingly appropriate, as all three e-books have long scrolling pages with a mixture of text and the widgets intertwined in them. Limiting the page length to that of a standard textbook, while still utilizing the widgets, could increase the book metaphor and thus improve the design and usability for the user (Wilson, Landoni, & Gibb, 2000). However, finding the appropriate balance of information for a specified page length may prove to be a difficult design task.

All three e-books have the design factors of font and color contrast between background and content in their top three highest ranked design heuristics. While all the fonts used among the e-books are sans-serif, they each use a different font. Font sizing is often specific to the zoom level of the user's web browser and the device they are accessing the e-book on. Because of this, it's difficult to properly design the font and

increase readability outside of setting a minimum font size. The colors used between all three of the e-books are also quite different, generally reflecting the producing company's colors. Despite the high average ranking of color contrast, some participants were not fond of the colors, particularly one of CS Circles primary colors, which is bright green. Participant feedback for color suggests that as long as the foreground is discernable from the background, the best choice of color is often specific to the user's person preference. A study by Chong, Lim, and Ling (2009) suggests using appropriate color contrast to increase readability, such as black text on a white background, and being consistent with font sizing.

The two highest ranked widgets by the participants in terms of usefulness were Active Code and Code Lens. There are the two widgets that display some form of code output. They allow more interaction and user creativity, through modifying and stepping about the code, compared to Multiple Choice widget. Parsons Problems is close to the user creativity level offered by Code Lens. However, there's a lack of detailed feedback and output from Parsons Problems that participants often requested in their suggestions. This is suggestive that interactivity coupled with detailed feedback are two key factors in designing an interactive widget that is both favorable and useable. The increased interactivity within the widgets is additionally beneficial for fostering creative learning, as supported by Hamed and Hosamn (2013).

5.4 Learnability

As expected, the participants scored low on the initial pretest. The pretest questions increased in difficulty, with the later questions requiring a level of coding knowledge that surpasses our ideal participant criteria for this study. Since only one of

the posttests had all its questions correctly answered, a ceiling effect did not occur for the study. Additionally, the same question was never missed twice, if one participant incorrectly answered a question, then the other participant would correctly answer it. This suggests that certain concepts within the e-book could use refinement to help better teach these missed concepts, as indicated by the incorrect questions.

The difference in percentage of correct responses for the pretest and posttest scores suggests that participants were learning, at least to some degree, from using the e-book. To further support this, the level of confidence in teaching the learned material the two participants reported were average and excellent. Having a level of confidence in teaching the material is indicative that the participants feel that they have actually learned and understand the material they interacted with in the e-book. However, with the small sample size for the study, it is difficult to make assumptions regarding the participants' learning.

The participants' consistency in reporting that the e-book would be beneficial for both teachers and students shows support for our claim of the e-book being an invaluable educational resource for teachers. As participants for the study were teachers, they were able to speak from experience regarding how beneficial the e-book could be for them. As suggested by one of the participants, the addition of a live chat feature could bring more educational benefits through increased discussion and active learning. More practice examples, as another participant suggested, could increase interactivity and thus student learning. Both of these suggestions support the use of interactivity within educational computer science e-books, as it is not only preferable to the user but enhances the learning process.

5.5 Limitations

This study had several limitations, the first being that it is primarily limited to its small sample size. Both studies are targeted toward teachers, who are quite busy during the school year, so enlisting their participation was quite difficult. The sequence for displaying the e-books and widgets to the participant always went in the same order of Runestone, Zyante, and then CS Circles. This repeated order could have caused a sequencing confound, particularly when identifying the purpose of the widgets. Finally, a few technical errors due to web servers occurred for participants when accessing the various e-books and widgets. Encountering such an error is likely to cause the participant to become frustrated or skip interacting with the widget, thus influencing their response.

5.6 Conclusion

This two part study analyzed the usability of interactive components within three educational interactive computer science e-books and examined the content knowledge gained from using our e-book as a teaching platform. Educational e-books with high levels of usability and interactivity improve the learning process, making them an effective alternative to current distance learning methods for teaching computer science. Results and participants' comments reveal that providing detailed feedback and maintaining a simple and clear design are key components for improving the usability of the interactive widgets and e-books reviewed in this study. The results also indicate that for these three e-books, the navigation and amount of information on the page need improvement, yet their use of color and font is well executed. While the context and topic of the question and topic are dependent, all four of the interactive widgets are

uniquely beneficial and their use is suggested. Additionally, the use of all three e-book platforms is suggested, with an equal preference toward Runestone and Zyante.

REFERENCES

- Al-Qassabi, H., & Al-Samarraie, H. (2013). Applying Gagne's Nine Events in the Design of an Interactive eBook to Learn 3D Animation.
- Baird, C., & Henninger, M. (2011). Serious play, serious problems: Issues with eBook applications. *Cosmopolitan Civil Societies: An Interdisciplinary Journal*,3(2), 1-17.
- Berg, S. A., Hoffmann, K., & Dawson, D. (2010). Not on the same page: undergraduates' information retrieval in electronic and print books. *The Journal of Academic Librarianship*, 36(6), 518-525.
- Campbell, A., Callaghan, G., McGarvie, D. W., & Hynd, M. Do students study and learn differently using e-Readers? A cross-discipline research investigation into the pedagogical implications of using e-Readers to study university level texts.
- Chong, P. F., Lim, Y. P., & Ling, S. W. (2009). On the Design Preferences for Ebooks. *IETE Technical Review (Medknow Publications & Media Pvt. Ltd.)*,26(3).
- Chowdhury, G.G.: Access and usability issues of scholarly electronic publications. In: Gorman, G.E., Rowland, F. (eds.) *International Yearbook of Library and Information Management, 2004/2005*, pp. 77–98. Facet Publishing, London (2004)
- Fenwick Jr, J. B., Kurtz, B. L., Meznar, P., Phillips, R., & Weidner, A. (2013, March). Developing a highly interactive ebook for CS instruction. In *Proceeding of the 44th ACM technical symposium on Computer science education* (pp. 135-140). ACM.
- Guzdial, Mark. (2013, December 10). *Providing CS Learning to High School Teachers and Others, Ubiquitously*. Retrieved from <http://home.cc.gatech.edu/csl/CSLearning4U>
- Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, 41(4), 393-416.
- Landoni, M., Wilson, R., & Gibb, F. (2000). From the Visual Book to the WEB Book: the importance of design. *Electronic Library, The*, 18(6), 407-419.
- Mioduser, D., Nachmias, R., Lahav, O., & Oren, A. (2000). Web-based learning environments: Current pedagogical and technological state. *Journal of research on computing in education*, 33(1), 55-76.

- Mishra, P., & Koehler, M. (2006). Technological pedagogical content knowledge: A framework for teacher knowledge. *The Teachers College Record*, 108(6), 1017-1054.
- Niess, M. L. (2005). Preparing teachers to teach science and mathematics with technology: Developing a technology pedagogical content knowledge. *Teaching and Teacher Education*, 21(5), 509-523.
- Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (2000). Innovating adaptive and neural systems instruction with interactive electronic books. *Proceedings of the IEEE*, 88(1), 81-95.
- Shiratuddin, N., Hassan, S., & Landoni, M. (2003). A Usability Study for Promoting eContent in Higher Education. *Educational Technology & Society*, 6(4), 112-124.
- Shulman, L. S. (1987). Knowledge and teaching: Foundations of the new reform. *Harvard educational review*, 57(1), 1-23.
- Stoop, J., Kreutzer, P., & Kircz, J. (2010). Reading and learning from screens versus print: a study in changing habits: Part 1—reading long information rich texts. *New Library World*, 114(7/8), 284-300.
- Torigoe, H. (2013, April). Developing an Effective E-Textbook for CS101 Students at UH Hilo: An iBook Instructional Module. Technology, Colleges, and Community (TCC) Worldwide Online Conference.
- Vikas, Y., Romanello, T., & Gramoll, K. (2000, June). Teaching dynamics online with only electronic media on laptop computers. In *Proceedings of the 2000 American Society for Engineering Education Annual Conference & Exposition (CD-ROM)*.
- Voogt, J., Fisser, P., Pareja Roblin, N., Tondeur, J., & van Braak, J. (2013). Technological pedagogical content knowledge—a review of the literature. *Journal of Computer Assisted Learning*, 29(2), 109-121.
- Wasecka, J. J. (2013). *Creating an E-book*. (Doctoral dissertation, State University of New York Institute of Technology).
- Wilson, R., Landoni, M., & Gibb, F. (2002). Guidelines for designing electronic books. In *Research and Advanced Technology for Digital Libraries* (pp. 47-60). Springer Berlin Heidelberg.

APPENDIX A

USABILITY SURVEY

Interactive Ebook Evaluation

4. What is the highest degree or level of school you have completed (or currently enrolled in)?

- High School
 Bachelor's Degree
 Master's Degree
 Post Graduate Degree
 Other (please specify)

5. Please rate your skill level in the following:

	1 (No Experience)	2	3	4	5 (Expert)
Programming (in any language)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Interface Design	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

6. Do you have six or more months of prior coding experience and knowledge?

- Yes
 No

7. Have you previously used an ebook?

- Yes
 No

8. Have you previously used an interactive ebook?

- Yes
 No

If yes, please specify which book(s):

9. Have you previously used an ebook to learn programming?

- Yes
 No

If yes, please specify which book:

Ebook Design

Interactive Ebook Evaluation

This survey is to evaluate designs for interactive ebooks to learn programming in Python. These ebooks use interactive widgets throughout the book to help users learn better. Please help us evaluate and compare the designs of these books.

Click on the links below and try out the designs. Please come back to this page to answer questions.

[Ebook Design 1](#)

[Ebook Design 2](#)

[Ebook Design 3](#)

10. Rate the **Ebook Design 1** on the following factors:

	1 (Poor)	2	3	4	5 (Excellent)
Navigation ability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Amount of information on the page	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Media arrangement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Page layout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Legibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
White space	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Color contrast between background and content	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please tell us, if you have any specific feedback

11. Rate the **Ebook Design 2** on the following factors:

	1 (Poor)	2	3	4	5 (Excellent)
Navigation ability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Amount of information on the page	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Media arrangement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Page layout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Legibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
White space	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Color contrast between background and content	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please tell us, if you have any specific feedback

Interactive Ebook Evaluation

12. Rate the **Ebook Design 3** on the following factors:

	1 (Poor)	2	3	4	5 (Excellent)
Navigation ability	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Amount of information on the page	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Media arrangement	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Page layout	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Font	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Legibility	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
White space	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Color contrast between background and content	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Please tell us, if you have any specific feedback

Design 1: Widget (1/4)

We will evaluate the first interactive widget in Design 1:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 1, Widget 1 : Activecode](#)

13. What do you think is the purpose of this widget?

14. Please describe what you think each button does.

Run	<input type="text"/>
Save	<input type="text"/>
Load	<input type="text"/>
Start Audio Tour	<input type="text"/>

15. Did you try to change some values in the code and run it?

- Yes. I tried to change the code.
- No. I realized I could change the code, but chose to not.
- No. I did not realize I could change the code.

Interactive Ebook Evaluation

16. If you encountered an error when running the code, was the provided feedback helpful?

- Yes
 No
 I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

17. Was there anything you found confusing or did not like about the widget?

Design 2: Widget (1/4)

We will evaluate the first interactive widget in Design 2:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 2, Widget 1 : Activecode](#)

18. What do you think is the purpose of this widget?

19. Please describe what you think each button does.

Run

20. Did you try to change some values in the code and run it?

- Yes. I tried to change the code.
 No. I realized I could change the code, but chose to not.
 No. I did not realize I could change the code.

Interactive Ebook Evaluation

21. If you encountered an error when running the code, was the provided feedback helpful?

- Yes
 No
 I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

22. Was there anything you found confusing or did not like about the widget?

Design 3: Widget (1/4)

We will evaluate the first interactive widget in Design 3:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 3, Widget 1 : Activecode](#)

23. What do you think is the purpose of this widget?

24. Please describe what you think each button does.

Run program	<input type="text"/>
Enter input	<input type="text"/>
Simple editor	<input type="text"/>
Visualize	<input type="text"/>

25. Did you try to change some values in the code and run it?

- Yes. I tried to change the code.
 No. I realized I could change the code, but chose to not.
 No. I did not realize I could change the code.

Interactive Ebook Evaluation

26. If you encountered an error when running the code, was the provided feedback helpful?

- Yes
 No
 I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

27. Was there anything you found confusing or did not like about the widget?

Design 1: Widget (2/4)

We will evaluate the second interactive widget in Design 1:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 1, Widget 2 : CodeLens](#)

28. What do you think is the purpose of this widget?

29. Please describe what you think each button does.

<< First	<input type="text"/>
< Back	<input type="text"/>
Forward >	<input type="text"/>
Last >>	<input type="text"/>

30. Please explain in brief, what you believe is the purpose of the red and green arrows.

31. Was there anything you found confusing or did not like about the widget?

Interactive Ebook Evaluation

Design 2: Widget (2/4)

We will evaluate the second interactive widget in Design 2:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 2, Widget 2 : CodeLens](#)

32. What do you think is the purpose of this widget?

33. Please describe what you think each button does.

Run program

34. Please explain in brief, what you believe is the purpose of the orange row highlighting.

35. Did you try to change the order of the code blocks and run it?

- Yes. I tried to change the order of the code blocks.
- No. I did not realize I could change the order of the code blocks.

36. Was there anything you found confusing or did not like about the widget?

Design 3: Widget (2/4)

We will evaluate the second interactive widget in Design 3:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 3, Widget 2 : CodeLens](#)

37. What do you think is the purpose of this widget?

Interactive Ebook Evaluation

38. Please describe what you think each button does.

<< First	<input type="text"/>
< Back	<input type="text"/>
Forward >	<input type="text"/>
Last >>	<input type="text"/>

39. Please explain in brief, what you believe is the purpose of the yellow and grey row highlighting.

40. Was there anything you found confusing or did not like about the widget?

Design 1: Widget (3/4)

We will evaluate the third interactive widget in Design 1:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 1, Widget 3 : Mixed up program](#)

41. What do you think is the purpose of this widget?

42. Were you able to figure out how to indent a code block, in this widget?

- Yes.
- Yes, but after a few attempts.
- No. I did not realize we could indent a code block.

Interactive Ebook Evaluation

43. If you encountered an error when interacting with the widget, was the provided feedback helpful?

- Yes
 No
 I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

44. Was there anything you found confusing or did not like about the widget?

Design 2: Widget (3/4)

We will evaluate the third interactive widget in Design 2:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 2, Widget 3 : Mixed up program](#)

45. What do you think is the purpose of this widget?

46. If you encountered an error when interacting with the widget, was the provided feedback helpful?

- Yes
 No
 I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

47. Was there anything you found confusing or did not like about the widget?

Interactive Ebook Evaluation

Design 3: Widget (3/4)

We will evaluate the third interactive widget in Design 3:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 3, Widget 3 : Mixed up program](#)

48. What do you think is the purpose of this widget?

49. Did you click for a hint?

- Yes
- No
- I did not realize I could

Other (please specify)

50. If you encountered an error when interacting with the widget, was the provided feedback helpful?

- Yes
- No
- I did not encounter an error

If yes or no, please specify why the feedback was or wasn't helpful:

51. Was there anything you found confusing or did not like about the widget?

Design 1: Widget (4/4)

We will evaluate the fourth interactive widget in Design 1:

Interactive Ebook Evaluation

Please click on the link below and try out the widget you see on the page.
Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 1, Widget 4 : Multiple choice questions](#)

52. Please describe what you think each button does.

Check Me

Compare Me

53. Was the provided feedback from correctly/incorrectly answering a question helpful to you?

- Yes
- No
- I did not notice the feedback

If yes or no, please specify why the feedback was or wasn't helpful:

54. Was there anything you found confusing or did not like about the widget?

Design 2: Widget (4/4)

We will evaluate the fourth interactive widget in Design 2:

Please click on the link below and try out the widget you see on the page.
Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 2, Widget 4 : Multiple choice questions](#)

55. Was the provided feedback from correctly/incorrectly answering a question helpful to you?

- Yes
- No
- I did not notice the feedback

If yes or no, please specify why the feedback was or wasn't helpful:

Interactive Ebook Evaluation

56. Was there anything you found confusing or did not like about the widget?

Design 3: Widget (4/4)

We will evaluate the fourth interactive widget in Design 3:

Please click on the link below and try out the widget you see on the page.

Once you have tried it out, please return to this survey to answer questions on that widget.

[Design 3, Widget 4: Multiple choice questions](#)

57. Please describe what you think each button does.

Check answer

58. Was the provided feedback from correctly/incorrectly answering a question helpful to you?

- Yes
- No
- I did not notice the feedback

If yes or no, please specify why the feedback was or wasn't helpful:

59. Was there anything you found confusing or did not like about the widget?

Compare Widget 1

Please review and compare the first widget of each design:

1. Activecode widget you tested in design 1. ([link](#))

Reference image below

Interactive Ebook Evaluation

Run

Save

Load

Start Audio Tour

```
1 for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "Paris"]:  
2     print("Hi", name, "Please come to my party on Saturday!")  
3
```

ActiveCode: 1 (ch03_4)

2. Activecode widget you tested in design 2. ([link](#))

Reference image below

Try 1.1.1: Executing Python code using the interpreter.

Click the play button to execute the code, and observe the output. Try changing the wage hours to 20 (half-time).

```
1 wage = 20 # Cre  
2 hours = 40 # Cre  
3 weeks = 50 # Cre  
4  
5 salary = wage * h  
6 print('Salary is:  
7 print(salary)  
8
```

```
wage = 20 # Create a variable 'wage' with value 20  
hours = 40 # Create a variable 'hours' with value 40  
weeks = 50 # Create a variable 'weeks' with value 50  
  
salary = wage * hours * weeks # Compute salary  
print('Salary is:', end=' ') # Print "Salary is:" on the screen  
print(salary) # Print salary on the screen
```

Copyright © 2013 Zyante Inc.

3. Activecode widget you tested in design 3. ([link](#))

Reference image below

Interactive Ebook Evaluation



60. Which design do you like better:

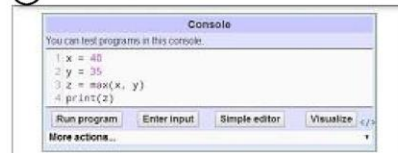
Ebook design 1



Ebook design 2



Ebook design 3



Please explain why:

Compare Widget 2

Please review and compare the second widget of each design:

Interactive Ebook Evaluation

CodeLens widget you tested in design 1. [\(link\)](#)

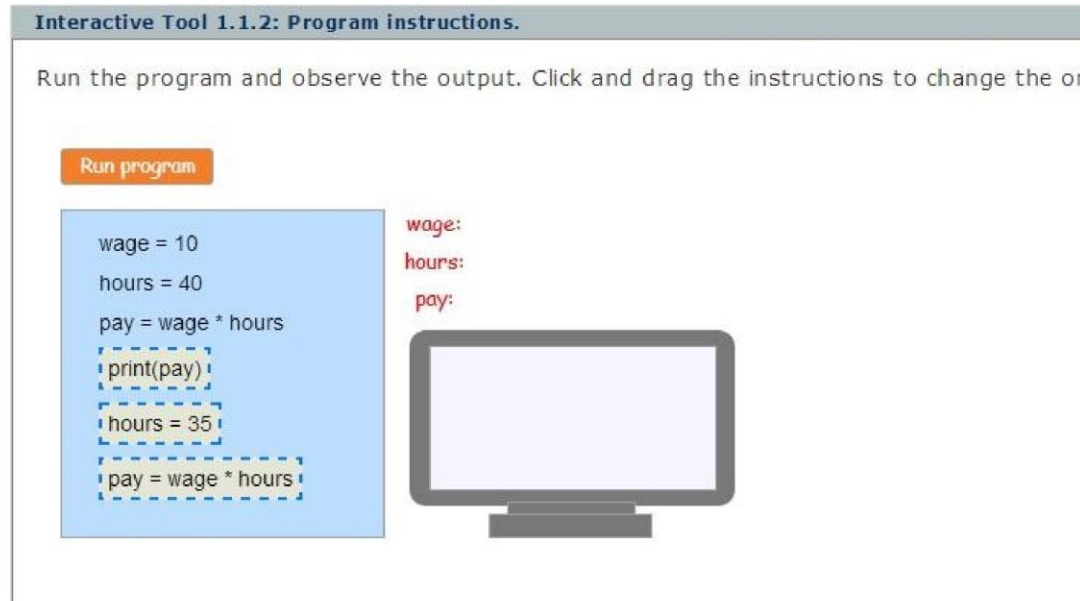
Reference image below



The screenshot shows a CodeLens widget with a code editor on the left and a 'Frames' panel on the right. The code editor contains two lines of Python code: `for name in ["Joe", "Amy", "Brad", "Angelina",` on line 1 and `print("Hi ", name, " Please come to my par` on line 2. A red arrow points to the start of line 1, and a green arrow points to the start of line 2. Below the code editor is a horizontal scrollbar and a vertical scrollbar. At the bottom of the widget are five buttons: '<< First' (grey), '< Back' (red), 'Step 1 of 15' (text), 'Forward >' (green), and 'Last >>' (grey). Below the buttons is a legend: a green arrow points to 'line that has just executed' and a red arrow points to 'next line to execute'. The text 'CodeLens: 1 (vtest)' is at the bottom right of the widget.

CodeLens widget you tested in design 2. [\(link\)](#)

Reference image below



The screenshot shows an interactive tool titled 'Interactive Tool 1.1.2: Program instructions.' Below the title is the instruction: 'Run the program and observe the output. Click and drag the instructions to change the or'. There is an orange 'Run program' button. To the left is a blue box containing a list of code instructions: `wage = 10`, `hours = 40`, `pay = wage * hours`, `print(pay)`, `hours = 35`, and `pay = wage * hours`. The last three lines are enclosed in dashed blue boxes. To the right of the code box are three labels: 'wage:', 'hours:', and 'pay:'. Below these labels is a computer monitor icon.

CodeLens widget you tested in design 3. [\(link\)](#)

Reference image below

Interactive Ebook Evaluation

[Computer Science Circles Homepage](#) | [Contact](#)

```
1 for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Thandi", "  
2     print("Hi ", name, " Please come to my party on Saturday!")
```

[Edit code](#)



<< First

< Back

Step 1 of 15

Forward >

Last >>

line that has just executed **next line to execute**

Program output:

Interactive Ebook Evaluation

61. Which design do you like better:

Ebook design 1



Ebook design 2



Ebook design 3



Please explain why:

Compare Widget 3

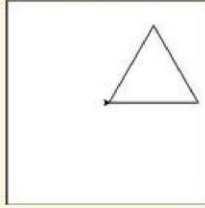
Please review and compare the third widget of each design:

Mixed up program widget you tested in design 1. ([link](#))

Reference image below

Interactive Ebook Evaluation

Q-93: The following program uses a turtle to draw a triangle as shown to the left, but up. The program should do all necessary set-up and create that, iterate (loop) 3 times, and each time through the loop go forward 175 pixels, and then turn left 120 degrees. After window to close when the user clicks in it.



Drag the blocks of statements from the left column to the right column and put them in the right order with the correct indentation. Click to see if you are right. You will be told if any of the lines are

or are incorrectly indented.

Drag from here

```
marie.forward(175)
```

```
wn.exitonclick()
```

```
marie.left(120)
```

```
# repeat 3 times  
for i in [0,1,2]:
```

```
wn = turtle.Screen()  
marie = turtle.Turtle()
```

```
import turtle
```

Drop blocks here

Check Me

Reset

Mixed up program: 1

Mixed up program widget you tested in design 2. ([link](#))

Reference image below

Interactive Ebook Evaluation

Question Set 1.1.3: Create an algorithm.

Imagine if you wanted to find every occurrence of the word "Voldemort" found in a stack of

DO the following for every paper P:

IF "Voldemort" found in the paper P:

FIND "Voldemort" in the paper P

RECORD the paper P

Drag and drop a term

Step 1

Step 2

Step 3

Step 4

Reset

Mixed up program widget you tested in design 3. ([link](#))

Reference image below

Scramble Exercise: Speed Calculator



You are in a bike race which goes up and down a hill. The grader will pre-define four variables for you: `uphillDistance` and `downhillDistance` give the distance (in km) of both parts of the race, and `uphillTime` and `downhillTime` give the time (in minutes) of how long it took you to complete each part of the race. Write a program that will print out your average speed (in km/min) for the entire race.

Drag and drop with your mouse to rearrange the lines.

[Click for a hint if you're stuck](#)

```
totalDistance = uphillDistance + downhillDistance
print(averageSpeed)
averageSpeed = totalDistance / totalTime
totalTime = uphillTime + downhillTime
```

Run program

</>

Interactive Ebook Evaluation

62. Which design do you like better:

Ebook design 1



Ebook design 2



Ebook design 3



Please explain why:

Compare Widget 4

Please review and compare the fourth widget of each design:

Multiple choice questions widget you tested in design 1. [\(link\)](#)

Reference image below

Interactive Ebook Evaluation

Q-103: Consider the following code:

```
import turtle
wn = turtle.Screen()
alex = turtle.Turtle()
alex.forward(150)
alex.left(90)
alex.forward(75)
```

What does the line "import turtle" do?

- a) It creates a new turtle object that can be used for drawing.
- b) It defines the module turtle which will allow you to create a Turtle object and draw with it.
- c) It makes the turtle draw half of a rectangle on the screen.
- d) Nothing, it is unnecessary.

Check Me

Compare Me

Multiple choice questions widget you tested in design 2. [\(link\)](#)

Reference image below

Question Set 1.1.4: Computer programming introduction.

Num	Question	Your answer	Result and e
1	A computer program consists of a series of instructions for a computer to execute.	<input type="radio"/> True <input type="radio"/> False	X
2	An instruction describes an action for a computer to carry out.	<input type="radio"/> True <input type="radio"/> False	X
3	The order of instructions in a program is important.	<input type="radio"/> True <input type="radio"/> False	X

Multiple choice questions widget you tested in design 3. [\(link\)](#)

Reference image below

Interactive Ebook Evaluation

Multiple Choice Exercise: Min and Max I ✓

What is the output of the following program?

```
x = 13
y = 7
a = max(x+y, x*2)
b = min(x, y)
print(a,b)
```

Your choice:

</>

63. Which design do you like better:

Ebook design 1

Q101 Consider the following code:

```
print('Hello')
a = 10
print(a)
print('Hello')
a = 20
print(a)
```

What does the line 'print(a)' do?

- It creates a new turtle object that can be used for drawing.
- It defines the window turtle which will allow you to create a Turtle object and draw with it.
- It prints the value that is stored in the variable 'a' on the screen.
- It prints 'a' if necessary.

Ebook design 2

Item	Question	Your answer	Result and explanation
1	A computer program consists of a series of instructions that a computer can execute.	<input type="checkbox"/> True <input checked="" type="checkbox"/> False	✗
2	An instruction describes an action for a computer to carry out.	<input type="checkbox"/> True <input checked="" type="checkbox"/> False	✗
3	The order of instructions in a program is important.	<input type="checkbox"/> True <input checked="" type="checkbox"/> False	✗

Ebook design 3

Multiple Choice Exercise: Min and Max I ✓

What is the output of the following program?

```
x = 13
y = 7
a = max(x+y, x*2)
b = min(x, y)
print(a,b)
```

Your choice:

</>

Please explain why:

ActiveCode

```
Run Save Load Start Audio Tour
```

```
1 for name in ["Joe", "Amy", "Brad", "Angelina", "Zuki", "Zhandi", "Paris"]:  
2   print("Hi, name, "Please come to my party on Saturday!")  
3
```

ActiveCode: 1 (ch03_4)



64. If you wanted to learn to program in Python, the Active Code widget would be useful:

Strongly Disagree Disagree Undecided Agree Strongly Agree

Active Code

If you have any suggestions, how might you improve this component to make it more useful?

Code Lens



65. If you wanted to learn to program in Python, the Code Lens widget would be useful:

Strongly Disagree Disagree Undecided Agree Strongly Agree

Code Lens

If you have any suggestions, how might you improve this component to make it more useful?

Mixed Up Program



66. If you wanted to learn to program in Python, the Mixed Up Program widget would be useful:

Strongly Disagree Disagree Undecided Agree Strongly Agree

Mixed Up Program

If you have any suggestions, how might you improve this component to make it more useful?

Multiple Choice Question

Q101 Consider the following code:

```
import turtle  
sh = Turtle(shape='s')  
sh.hideturtle()  
sh.penup()  
sh.goto(100,100)  
sh.pendown()
```

What does the line "import turtle" do?

- It is a module in the turtle object that can be used for drawing.
- It defines the turtle shape which will allow you to create a Turtle object and draw with it.
- It makes the turtle draw half of a rectangle on the screen.
- It nothing, it is unnecessary.

Question 101 of 100 Computer programming introduction

Question	Answer	Correct	Explanation
1. A computer program consists of a series of instructions for a computer to execute.	<input type="radio"/> True <input type="radio"/> False	<input checked="" type="checkbox"/>	
2. An algorithm describes an action for a computer to carry out.	<input type="radio"/> True <input type="radio"/> False	<input checked="" type="checkbox"/>	
3. The order of instructions in an algorithm is important.	<input type="radio"/> True <input type="radio"/> False	<input checked="" type="checkbox"/>	

Multiple Choice Exercise: Min and Max 1

What is the output of the following program?

```
x = 13  
y = 7  
z = max(x+y, x*2)  
b = min(x, y)  
print(a,b)
```

Your choice:

67. If you wanted to learn to program in Python, the Multiple Choice Question widget would be useful:

Strongly Disagree Disagree Undecided Agree Strongly Agree

Multiple Choice Question

If you have any suggestions, how might you improve this component to make it more useful?

APPENDIX B

LEARNABILITY PRETEST

4. What is your ethnicity? (Please select all that apply.)

- American Indian or Alaskan Native
- Asian or Pacific Islander
- Black or African American
- Hispanic or Latino
- White / Caucasian
- Prefer not to answer

5. Please select all previous programming languages with which you have experience.

- None
- Alice
- Basic
- C
- C++
- Cobol
- Fortran
- Java
- Java Script
- Jeroo
- LightBot
- Lisp
- LOGO
- Pascal
- Perl
- Processing
- Python
- Ruby
- Scheme
- Scratch
- Smalltalk
- Visual Basic

Other (please specify)

6. How would you rate your knowledge of programming?

Extremely Poor Below Average Average Above Average Excellent None

Programming Knowledge

Other (please specify)

7. How much programming experience do you have? If none, please enter 'None.'

8. How much, if any, of your programming experience is with Python? If none, please enter 'None.'

9. Have you read or worked through any Python books on your own?

Yes
 No

Python Pre-Test

To give us a sense of your prior understanding of Python before starting to use the book, please answer these questions. If you do not know the answer, please enter "I do not know" in the text box.

```
marks = 200 + 250
euros = marks / 5.0
```

10. What will be the value in marks and euros after the above lines of code execute?

marks:

euros:

```
denominator = 4
if denominator == 0:
    print 'Mere mortals cannot divide by zero.'
else:
    print 1000 / denominator
```

11. What will the above program print when it's run?

```
def check_info(name, age):
    return len(name) >= 0 and age >= 0

def print_info(name, age):
    if not check_info(name, age):
        return False
    print 'First name:', name
    print 'Age (years):', age
    return True

if print_info('John',15):
    print print_info('Paul',-1000)
else:
    print 'Better luck next time'
```

12. What will the above program print when it's run?

```
first = [10,5,10,6]
print first[3]
second=[3,1,-2]
print second
second[2] = second[2] + 1
print second[2]
```

13. If the code above was executed, three things would print. What would they be?

1.
2.
3.


```
first = [10,5,0]
first[1] = -5
value = first[2]
print first
second = [3,1,3,value]
third = second
second[3] = 100
print second
```

14. If the code above was executed, two things would print. What would they be?

1.

2.

APPENDIX C

LEARNABILITY CHAPTERS 1 & 2 POSTEST

3. What is the value of `var3` after the following code is executed?

```
var1 = "cat "  
var2 = "dog "  
var3 = "bird "  
var1 = var2  
var3 = var1  
var2 = "fish "
```

- dog
- fish
- cat
- bird

4. A named space that can hold a value is a ____?

- Variable
- Turtle
- Method
- Procedure

5. The kind of data which can be letters, digits, and other characters, usually delimited by quotes, is a ____?

- Integer
- Float
- String
- Double

6. Which feature(s) of these chapters did you find to be the most helpful?

7. What would you change, if anything, to make the readings more effective for you?

8. Do you feel like this would be an effective and beneficial tool for students? teachers?

APPENDIX D

LEARNABILITY CHAPTERS 3 & 4 POSTEST

3. What is printed after the following code is run?

```
Name = "John Smith"  
Words = " likes to play outside."  
Print (Name + Words)
```

- NameWords
- Name + Words
- John Smith + likes to play outside.
- John Smith likes to play outside.

4. In the following code, what does the “=” achieve?

```
Cat = "meow"
```

- Tests for equivalence
- Assigns the name on the left to the value on the right
- Moves the value from the right to the left
- Creates a new object

5. Which of the following line(s) of code will print out "My name is Morrissey".

- ```
print(My name is Morrissey)
```
- ```
var1 = "My name is "  
var2 = "Morrissey "  
var3 = var1 + var2  
print(var3)
```
- ```
M = "M "
orrissey = "orrissey "
print("My name is " + Morrissey)
```
- All of the above

**6. Which feature(s) of these chapters did you find to be the most helpful?**

**7. What would you change, if anything, to make the readings more effective for you?**

**8. Do you feel like this would be an effective and beneficial tool for students? teachers?**

## APPENDIX E

### LEARNABILITY CHAPTERS 5 & 6 POSTEST

**3. What value is printed when the following code is executed?**

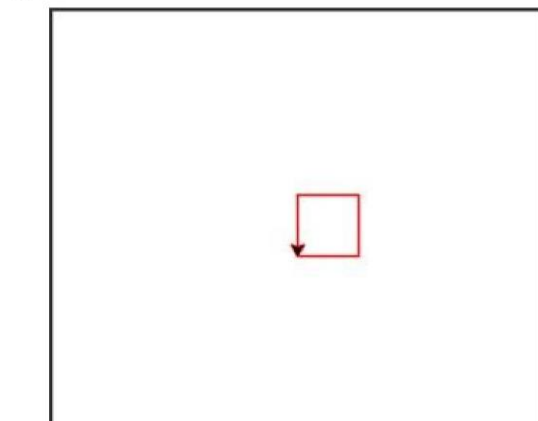
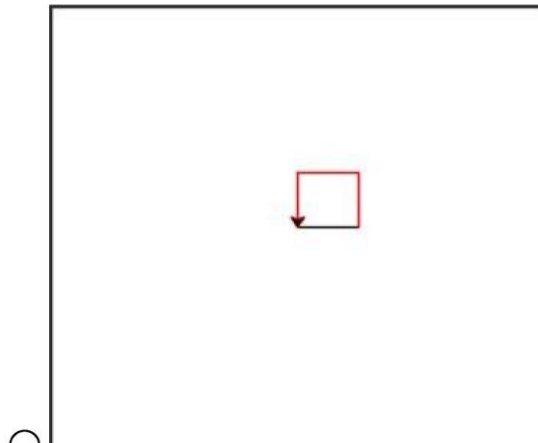
```
def myFunction(Parameter):
 Value = "First"
 Value = Parameter
 print (Value)
```

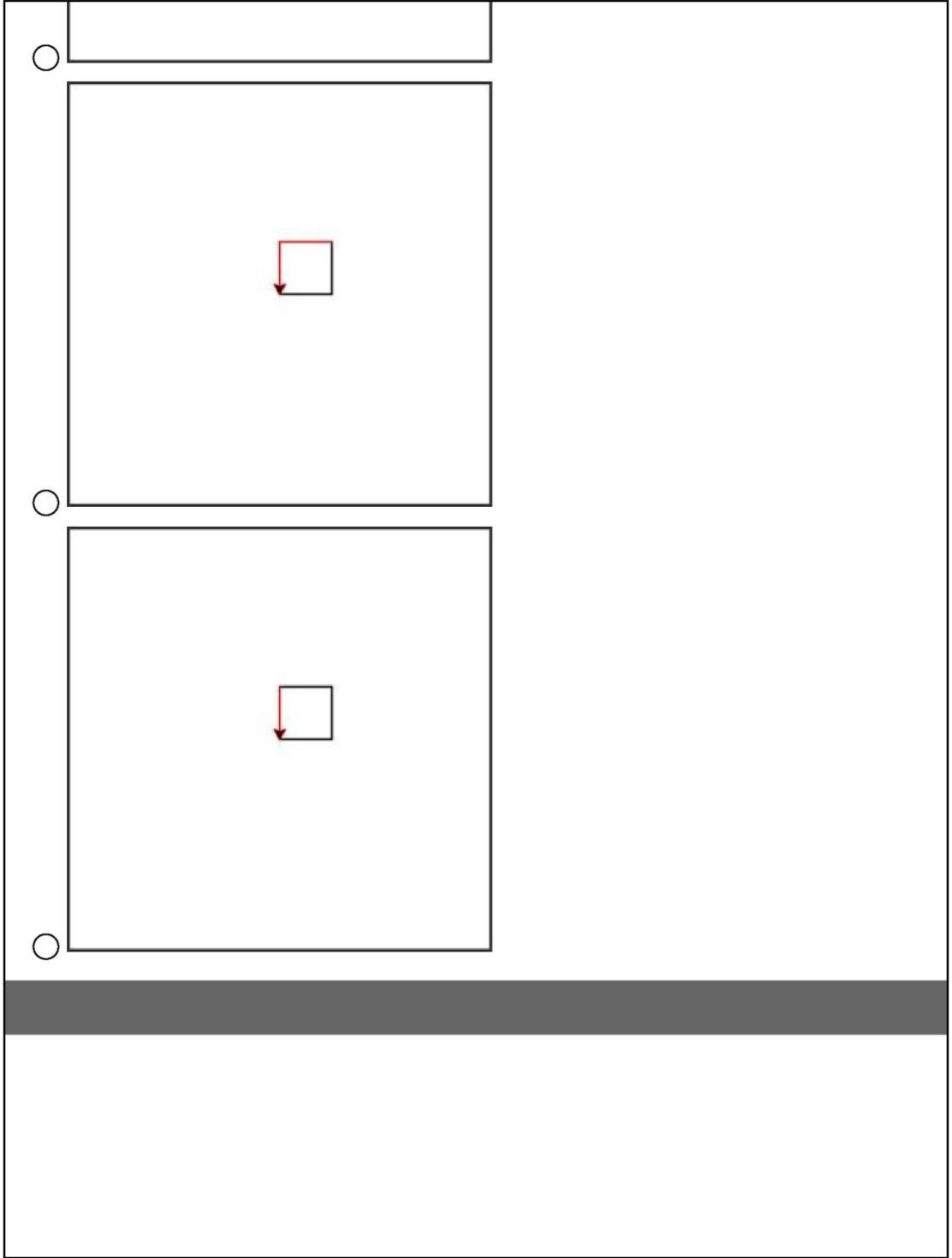
```
myFunction("Second")
```

- Value
- Second
- Parameter
- First

**4. Which square would the following code produce?**

```
from turtle import *
screen = Screen()
alice = Turtle()
alice.forward(50)
alice.left(90)
alice.forward(50)
alice.left(90)
alice.pencolor("red")
alice.forward(50)
alice.left(90)
alice.forward(50)
```







**5. Given the following lines of code, which will move the turtle Bob 50 units forward?**

```
from turtle import *
space = Screen()
Bob = Turtle()
```

- Bob.move(50)
- Bob.left(50)
- Bob.forward(50)
- Bob.right(50)

**6. A named sequence of statements that returns a result is known as a \_\_\_?**

- definition
- procedure
- turtle
- function

**7. Which feature(s) of these chapters did you find to be the most helpful?**

**8. What would you change, if anything, to make the readings more effective for you?**

**9. Do you feel like this would be an effective and beneficial tool for students? teachers?**

## APPENDIX F

### LEARNABILITY CHAPTERS 7 & 8 POSTEST

**3. What is the last line to be printed when the following code is run?**

```
Number = 0
while Number <= 10:
 Print ('Number: ', Number)
 Number = Number + 1
```

- Number: 10
- Number: Number
- Number: 0
- Number: 11

**4. When the following code is run, how many times is "hello" printed?**

```
helloArray = [1, 2, 3]
for x in helloArray:
 print "hello"
```

- 1
- 2
- 3
- 4

**5. What will occur when the variable `number` is set to the value of 5?**

```
number = 5
while number <= 5:
 if number < 5:
 number = number + 1
 print(number)
```

- The while loop will loop indefinitely
- The value of `number` will be printed exactly 1 time
- The while loop will never get executed
- The value of `number` will be printed exactly 5 times

**6. Which feature(s) of these chapters did you find to be the most helpful?**

**7. What would you change, if anything, to make the readings more effective for you?**

**8. Do you feel like this would be an effective and beneficial tool for students? teachers?**

**9. How much confidence do you have in teaching the material you've read and learned from chapters 1-8 of the ebook?**

Extremely Poor

Below Average

Average

Above Average

Excellent

**10. Is there anything else we could provide you with that you feel would be beneficial to your learning process?**