WELCOME TO COMPUTER SCIENCE:

DESIGNING A COMIC TOUR OF COMPUTERS AND

COMPUTING

by

AUDRA MCNAMEE

A THESIS

Presented to the Department of Computer and Information Science
and the Robert D. Clark Honors College
in partial fulfillment of the requirements for the degree of
Bachelor of Science

June 2022

An Abstract of the Thesis of

Audra McNamee for the degree of Bachelor of Science
in the Department of Computer and Information Science to be taken June 2022

Title: Welcome to Computer Science: Designing a Comic Tour of Computers and
Computing

Approved: _____*Dr. Kathleen Freeman*_____

Primary Thesis Advisor

While the number of high-quality educational comics is growing, there are no modern long-form comics discussing computer science at an undergraduate level. The computer science comics that do exist, along with being for a younger audience, are generally focused on teaching the reader programming concepts without exploring other aspects of computer science. For this thesis I scripted and drew the 56-page comic *Welcome to Computer Science*, which introduces the reader to computer science concepts including computer architecture, programming languages, and the internet. As a narrative comic written for an undergraduate audience, it can draw in readers who otherwise might not choose to engage with the material. As a breadth-first introduction, the comic provides the reader with a foundational understanding of computers and computer science; this work may provide even more experienced students with a better understanding of how their computer science classes relate to the rest of the field.

Acknowledgements

Table of Contents

List of Figures

Literature Review

Computer scientist, teacher, and comics creator Gene Luen Yang (2018) proposes that comics are not merely an effective teaching tool, but the most natural format for communicating complicated concepts. He argues this is because students have grown up in a visual world, and so comics are a way to present material which more completely taps into the way students tend to process information. Other comics creators have their own theories as to why comics are an effective method for scientific education. Nuclear engineer and comics writer Jim Ottaviani (2011b) points out that science has always used diagrams and visualizations to convey meaning and claims comics as a natural extension of this tradition. Ottaviani also says the fact that comics typically have narrative structure is key, because "stories are what make learning worthwhile". These are common explanations for why comics are an excellent format to use for science communication. Collver & Weitkamp (2018) interviewed science comic creators and grouped the major strengths the creators saw in their form into four categories: First, the visual aspect of comics grabbed attention, and was beneficial in illustrating complicated scientific concepts. Second, the narrative of comics attracted readers who like stories, making the science relatable and memorable to the reader. Third, the reader has control over the speed at which information is presented when learning from a comic. In video, the other major visual form, the speed of information presented is determined by the video's creator. When watching a video, to spend more time on an explanation or to move on from information already known a student must skip through the video. When learning from a comic, readers can spend longer on

complex sections, skim information they already understand, and return to earlier sections without breaking the flow of information presentation. Fourth, the still existent perception of comics as "not difficult," based on stereotypes of comics as unsubstantial material for kids, has the advantage of ensuring that science comics are not intimidating. Ideally, this means people hesitant to pick up scientific articles or textbooks, perceiving them to be uninteresting or an imposing commitment, would remain willing to read science comics.

Another form of educational comics is the emerging field of "data comics," or the use of comics to present scientific data. Bach et al. (2017) define a data comic as a comic where the narrative is formed from data and aims to explain this data in an accessible manner. Data comics have four beneficial components: Comic drawings turn abstract data into concrete images, ideally visualizations that readers without prior knowledge on the subject can understand. Comic pages break complicated information into smaller chunks (panels), creatively guiding the reader through a topic. Comics can present data within a narrative, including a fictional narrative, to tell a story. Finally, data comics integrate words and images, which is important because both mediums have strengths and weaknesses. Each of these four components must be carefully balanced to convey information clearly. Leaning too heavily into the technical "data" or the nonfactual "art" muddies the message the data comic is attempting to clarify. Data comics are a powerful tool for outreach, though Bach et al. mention that there are a huge number of questions still to explore, including what makes an effective data comic and how the form can be varied.

Clearly, educational comics about scientific topics is an expanding and promising field. Coming with that promise, though, are tensions and areas of ill-definition. Though comics have the potential to draw in readers who may not otherwise engage with science, and to give readers a different experience with scientific information, it can be difficult for science comics to reach readers. Many comics readers do not expect science comics, and many scientifically minded people do not see comics as scientific enough. Neuroscientist and comics creator Matteo Farinella (2018) believes this perception of comics is what has led to the relative dearth of studies researching the way that science comics are read and understood. When studies about learning do include comics, the comics are usually included as a supplemental reading tool rather than the focus of the study. While this research is important, the way that readers process scientific information through comics remains largely unknown. The small scale of existing research cannot clarify whether the learning benefits from comics are a result of their novelty, their visualizations, from the fact that they often have a narrative structure, some combination of the above, or other aspects entirely. Also, current research does not have the capacity to consider the great variability within comics: differences of artistic style, narrative structure, and information presentation.

Despite this uncertainty, Farinella identifies several aspects of comics that may be particularly beneficial in science communication: first, they appear to be most useful for students without prior information, so it is possible they could be a good way to give the reader a background in a topic before moving on to information formatted into other ways. Visualizations broken into panels, as Bach et al. (2017) argue, could be especially good at explaining complex information in an unintimidating way. Narratives have been

found to be easier to remember, and while not all science comics use fictional narratives to structure their story, they use narrative at a higher rate than other modes of scientific communication. Finally, science communication often relies on metaphor, something that comics are particularly adept at illustrating (Farinella, 2018). Science comics are a subject with few rules and fewer guidelines, but a huge amount of creative space and potential.

Turning for guidance to the comics and graphic novels themselves, it is evident that there are many effective science comics. Notable long-form examples include: The graphic novel *Logicomix: An Epic Search for Truth* (Doxiadēs et al., 2009), which uses the narrative of logician Bertrand Russell's life to teach the reader about set theory. *Neurocomic* (Farinella & Roš, 2014) uses the narrative of a man falling into his crush's brain to teach the reader how the brain works. *BrainComix* (Marmion & Monsieur B., 2021) is a fictional interview with the brain, and *The Body Factory* (Chochois, 2021) is about prosthetics and human augmentation. *Howtoons: Tools of Mass Construction* (Griffith et al., 2014), a science fiction story about two siblings in the future who go on adventures and solving problems by creating gadgets. The book teaches the reader how to make these gadgets, everything from useful knots to marshmallow shooters, and uses physics and chemistry to explain how the gadgets function. *Clan Apis* (Hosler, 2013) and *Last of the Sandwalkers* (Hosler, 2015) are narrative stories whose main characters are insects. These books contain accurate information about the habits of these insects, and emotional narratives about their lives. *Feynman* (Ottaviani & Myrick, 2011a) tells the life story of physicist Richard Feynman, including information about his breakthroughs in physics, and his approach to solving hard problems; other books

Ottaviani has written about scientists include *Hawking* (2019)*, Primates: The Fearless Science of Jane Goodall, Dian Fossey, and Biruté Galdikas* (Ottaviani & Wicks, 2013)*,* and *The Imitation Game: Alan Turing Decoded* (Ottaviani & Purvis, 2016)*.* Three books focused on computer science are *The Thrilling Adventures of Lovelace and Babbage: The (Mostly) True Story of the First Computer* (Padua, 2015), *Secret Coders* (Yang & Holmes, 2015), and *The Cartoon Guide to Computer Science* (Gonick, 1983). *The Thrilling Adventures of Lovelace and Babbage* chronicles the fictional adventures of real-life computer pioneers Ada Lovelace and Charles Babbage and is more focused on humor than on education. *Secret Coders,* written for a middle-grade audience, is about a group of friends going on adventures and solving puzzles by applying computer science principles. *The Cartoon Guide to Computer Science* is an introduction to computer science from 1983. It is not focused on a consistent cast of characters, one element that is helpful for building narrative and reader attention, and it is currently out of print. Works written for adults about computer science with an illustration component include the works of Amy Wibowo, with her BubbleSort Zines project and Julia Evans, with her Wizard Zines project. Both these collections of zines introduce readers to topics like how to build your own calculator, or how to use Linux debugging tools. These zines are prose with some illustrations, rather than comics. Taken together, these books are aimed at a diverse set of readers, and differ in how directly they explain scientific concepts, as well as in how much science they are trying to teach the reader. What they have in common is a goal of encouraging curiosity and wonder about the science on which they are focused.

The fact that visual metaphors are so effective in comics, and the way the reader can pause and examine complex pages for as long as they would like, makes comics a natural fit for computer science education. Indeed, there are many examples of using comics and comic-like objects for this purpose. Many of them, like *The Magic School Bus Gets Programmed* (Cole & Sykora, 1999), *Hello Ruby: Adventures in coding* (Liukas, 2015), and *Ara the Star Engineer* (Singh & Konak, 2018), are not comics, but rather illustrated picture and chapter-books. *Secret Coders* (Yang & Holmes, 2015) is a narrative comic. What these all have in common is they are aimed at a younger, elementary or middle school, audience.

For a specifically undergraduate audience, Suh et al. (2020) have incorporated comic strips ("coding strips") into introductory CS lectures and examined their impact on computer science learning. This work is based in the educational theories of concreteness fading and dual coding. Concreteness fading is a teaching technique where an idea is illustrated at various levels of abstraction in order to show the relationship between those levels of abstraction. In this case, code is the abstract idea, and the comics are the concrete illustration of a computer science concept (Suh et al., 2020). Dual coding refers to the fact that people take in information verbally and visually and so comics, which have both text and images, may activate both channels leading to better retention of the information (Suh et al., 2021). Suh et al. report that in practice students find "coding strips" to be effective and enjoyable additions to lecture, except when asked to interpret them specifically as code. This work is the most prominent current incorporation of comics and undergraduate-level computer science education. However, the coding strips are supplemental material and short; many of them are

single-panel images. While the researchers ensure they have a narrative within the coding strips, due to the nature of the one-off strips characters are created and vanish quickly. This does not provide space to create complex stories, eliminating one of the possible beneficial dimensions of comics. These strips are a novel method of undergraduate computer science education, but they are not taking advantage of all the possibilities comics can offer.

Looking at computer science education best practices, Pears et al. (2007) put forward four major areas to consider when teaching an introductory computer science course. The first is the curricula, which should be based on commonly accepted standards and fit with the courses that will follow in the sequence. The second is pedagogy—the metaphors used to describe what a computer is and how it works should be considered thoughtfully, and the aspect of computer science that the course is focused on should be clear to the instructor. Thirdly, the programming language or languages chosen for the course must strike a balance between the often-unwieldy languages used in industry, and the languages made for teaching which will not be useful to students going forwards. Lastly, the tools (integrated development environments, visualization software, and other ways of interacting with the computer and seeing what it is doing) should be selected with care. Vakil (2018) argues that another important component of this education, especially as computer science becomes a subject that is offered or even required of every student, is a robust discussion on ethics and the way that computer systems impact the world. Her first critique of computer science education is that "linking the need for CS in schools to the interests of multinational corporations obscures the sociopolitical implications, relevance, and,

ultimately, liberatory possibilities of teaching and learning CS" (Vakil, 2018). She says

that increasing equity in computer science education should not be solely focused on

preparing students traditionally underrepresented in the field for jobs at large tech

companies. Instead focus should be on designing curricula that "engag[es]…ethical and

political implications as well as unrealized possibilities for technology to transform and

empower communities" (Vakil, 2018).

The computer science comics identified previously are focused on teaching

"computational thinking" and, to a lesser extent, teaching programming syntax. While

this is valuable and important information, it is only one part of computer science, and

possibly the part of computer science that is best understood by non-computer

scientists. There are other philosophies of teaching computer science, one of which is

the focus of *Invitation to Computer Science* (8th edition):

> [A] breadth-first computer science service course would … cover
> foundational issues such as algorithms, abstraction, hardware, computer
> organization, system software, language models, and the social and
> ethical issues of computing. An introduction to these core ideas exposes
> students to the overall richness and beauty of the field and allows them
> not only to use computers and software effectively, but also to
> understand and appreciate the basic ideas underlying the discipline of
> computer science and the creation of computational artifacts …
> [S]tudents who complete such a course will have a much better idea of
> what a major or a minor in computer science will entail (Schneider &
> Gersting, 2019).

This breadth-first approach has the strength of going from concrete, real machinery to

more complex and abstracted higher-level code. The concrete nature of this method of

explaining computers makes it particularly adaptable to the comic page. It is also

particularly suited to the kind of comic reader that science comics tend to satisfy the

most—beginners in computer science who may not feel willing to commit to reading a

textbook or other intimidating book about computer science. Ideally, a science comic which focuses on the breadth of computer science would provide a framework with which to process computer science information, making the process of learning more computer science down the line less confusing.

For this thesis I scripted and drew the 56-page comic *Welcome to Computer Science*, represented in full in the Appendix, which introduces the reader to computer science concepts including computer architecture, programming languages, and the internet. *Welcome to Computer Science* is, to my knowledge, unique as an undergraduate-level comic which introduces the breadth of computer science.

Discussion

Topic Selection

The "breadth" of computer science is an open-ended guiding principle. The first stage of the writing process required I select specific topics. Several informal criteria guided my topic selection: how fundamental the topic was to understanding computers, how capable I felt in explaining it, and whether it could fit in some useful form in eight drawn pages. This last criterion was the most constraining.

I chose to cover computer architecture, programming languages, and the internet. These topics offer a natural progression from the nuts and bolts of the computer, through low-level encoding of information, to the higher-level ways that people code on computers and use computers to interact with each other. Each of these topics are pertinent to a person who wants to learn computer science, and, I would argue, anyone who uses computers. These topics also build naturally in complexity.

I chose computer architecture for the first section because, as I say in the comic, computer science is not solely about computers, and yet, it is difficult to do computer science without them. I chose programming languages for the second section because often a person's first interaction with computer science is through learning a programming language. Though that is not what this comic is focused on, the comic does define what a programming language is and explains generally how programming languages work. I chose the internet for the third section because the internet is almost certainly the primary use of the reader's computer, so it makes sense for them to learn how it works both on an algorithmic and an infrastructural level.

15

I initially planned for a fourth section about artificial intelligence and computer ethics, presenting several of the prominent computer-related issues of the moment, but this section remained ill-defined and too conceptually large to fit within eight concise pages.

I used reference books to clarify my knowledge of the topics I was writing about, and to make sure that the metaphors I was using were relatively standard. I considered trying to create novel metaphors, but many computer science education cliches (like the internet being akin to the postal system) are widely used because they are apt. I also liked the ability to fully illustrate metaphors that usually are described with words—drawing out an IP packet as an actual postcard, visualizing the travel of information across the world, and so on. I feel the comics format made me able to add more life to the standard computer metaphors.

Scripting

After deciding the topics on which I would be focusing, I consulted reference books, as well as my own knowledge of the topics, to decide the key points that I wanted to emphasize for the reader in each section.

Things I kept in mind as I was creating this project included:

1) Was I using *comics* to deliver computer science information, or were the images incidental to the text?

2) What assumptions was I making about the knowledge of my audience?

3) How well did the sections flow together?

Figuring out what information was inessential to the project and could be removed was, as always, the most important step.

Computer science is replete with jargon, and I was careful to minimize its use in any case where it came up. While it is useful to know technical language when working in the field, teaching this language was not one of the goals of my work. To keep *Welcome to Computer Science* accessible to readers who didn't know these technical terms, I avoided using jargon when possible and used alternate phrasing alongside jargon everywhere else. The subjects of each of the comic's sections, "computer architecture," "programming languages," and "the internet" do not inherently inspire interest. To address this, I wrote the title of each section in two ways—on the title page of the section, I rephrased the content of the section alternately with less-jargony synonyms, a question, or an observation. On the following page which faced the first page of the section and otherwise would have been blank I wrote the technical term for the topic at hand.

The thesis title was another struggle. "Computer science" itself is a term that does not necessarily have an intuitive meaning to someone outside of the field, nor does it have comprehensible synonyms. Additionally, the closest inspirations for my work have names like *Logicomix, Neurocomic, BrainComix,* or *The Cartoon Guide to Computer Science*, none of which were inspiring to me. I wanted the title to convey that the comic would be about computer science, would be accessible to complete beginners, was for a high school/undergraduate/adult audience, and that it took a breadth-first approach rather than being focused on programming. I also potentially wanted to indicate that it was a comic. I settled on *Welcome to Computer Science,* uncomfortably

close to *Invitation to Computer Science,* but I appreciated how it worked on the cover, juxtaposed with an image of the Narrator opening the door to the space where the reader would be spending the bulk of the comic.

The last part of writing this *Welcome to Computer Science* was creating the Further Reading pages for each section. Because my goal with the comic was to cover a lot of material in an accessible way, I prioritized ease of understanding over going deep into a topic. This made the Further Reading section essential—because it was here that I could provide a way to go deeper into the topics I mentioned, in the (ideal) situation that a reader would be so interested in a topic I had described that they would want to learn more. I decided to format the Further Reading by page number, essentially providing endnotes on each page. The essential aspects of the Further Reading page included:

1) The formal names for every topic introduced in the comic.

2) Any kind of off-beat resource (comics, videogames, interactive websites, and short videos) I personally enjoy for learning more about a topic.

The final Further Reading section (after the Conclusion) also references the three major books that I referenced when writing the comic, in case readers are interested in learning from traditional textbooks. Internet searches can also bring up useful information on each topic within the comic, if that is the route a reader chooses.

Choosing the "narrative frame"

Most educational comics are not a straightforward collection of illustrated facts. Instead, they use a (fiction or nonfiction) narrative frame in conjunction with educational material to create a richer reading experience.

There are several narrative framing methods that educational comics use. The first one is to focus on an audience surrogate character who, like the audience, doesn't know anything about the topic at hand and, for fictional reasons, must learn about it. Oftentimes this is accomplished by interacting with one or more other fictional characters who do have expertise and who teach them; the key aspect of this framing device is that the audience surrogate is learning with the audience. This framing device is used in *Neurocomic* (Farinella & Roš, 2014), *BrainComix* (Marmion & Monsieur B., 2021), *Secret Coders* (Yang & Holmes, 2015), and *The Body Factory* (Chochois, 2021).

The audience surrogate method naturally heightens the emotions of the piece. The character the audience is meant to relate to is put through fictional narrative, adding drama to the piece. They're also engaging with the material and expressing emotions the audience might also be feeling, alternately perplexed, bored, and intrigued. This method tends to work well with a more fantastic method of teaching about the subject, by: taking the audience surrogate literally into the subject at hand (for example, traveling into the brain in *Neurocomic*); by creating a narrative where the subject matter is personally relevant to the audience surrogate, and pulls them through a story (the main character undergoing a traumatic injury and learning about prosthesis in *The Body Factory,* or the school-age kids puzzling out mysteries with code in *Secret Coders*); or having an absurdly fantastic setup (like interviewing the brain and other organs in a talk-show situation in *BrainComix*). This framing device lends itself to creating a narrative out of the teaching experience, useful because people are narrative creatures. Recall that narratives are one of the elements comics creators identified as being particularly beneficial about educational comics (Bach et al., 2017).

19

However, this narrative frame is not universally applicable. It works better when the topic would naturally draw a character into it in a compelling way. In *The Body Factory*, the educational narrative covers "the first prosthetics to the augmented human." Likewise, the emotional narrative follows the audience stand-in getting into a motorcycle crash and losing one of his legs, and his coming to terms with the loss, and becoming comfortable with his prosthesis. In *Secret Coders,* the main characters are directly solving suspenseful puzzles with programming principles, so the framing device adds energy and excitement the comic would not otherwise have. Other comics that use this method can feel more artificial, and instead of bringing the reader closer to the narrative can drive them away. *Neurocomic* is a story in which the audience surrogate sees a beautiful woman from afar, is attracted to her, and then is sucked into the reader's brain and must work his way through the reader's various brain structures to find her again. I like the whimsy of the way the brain is drawn, but fundamentally the main character's plight is more alienating than relatable. Partially this is because the only woman with a speaking role in the comic is relegated to object of attraction for the audience surrogate to chase with the help of several neuroscientists, but also this is because the audience surrogate is an unwilling student; he has been forced into a situation against his will, and while he begins feeling wonder at the brain's structures, the knowledge he has to learn to escape and move on to his happy ending is entirely separate from his personal motivation to talk to the woman. Likewise, the talk-show frame of *BrainComix* falls flat in practice. While the question-response structure of the interview between the talk-show host and the brain makes sense, the format of a talk show does not lend itself naturally to comics (without increasingly whimsical situations

it would devolve into talking heads) and the talk show itself does not have any thematic meaning in the comic, instead simply adding noise into the comic. If the audience surrogate's journey and the educational narrative do not smoothly parallel, they can feel like the bumpy product of a lack of trust in the reader's capacity to pay attention without a requisite number of one-liners and crude but irrelevant jokes. Or the two narratives can directly clash, resulting in the fictional component distracting the reader from the educational component, and/or the educational component causing the fictional component to ring false.

The audience surrogate technique was the framing device that I initially considered, planning on centering the comic around a student perplexed in a computer lab, and a sentient computer who answers their questions. I like the fact that this would make the comic into a conversation between two characters. However, as I began scripting this comic, the idea started feeling like it would be unwieldy and distracting from the educational points I was trying to make. While I believe that this type of educational story is powerful, without a standout idea for an audience surrogate narrative, I chose to not use this framing technique. The risks outweighed the potential benefits for this project.
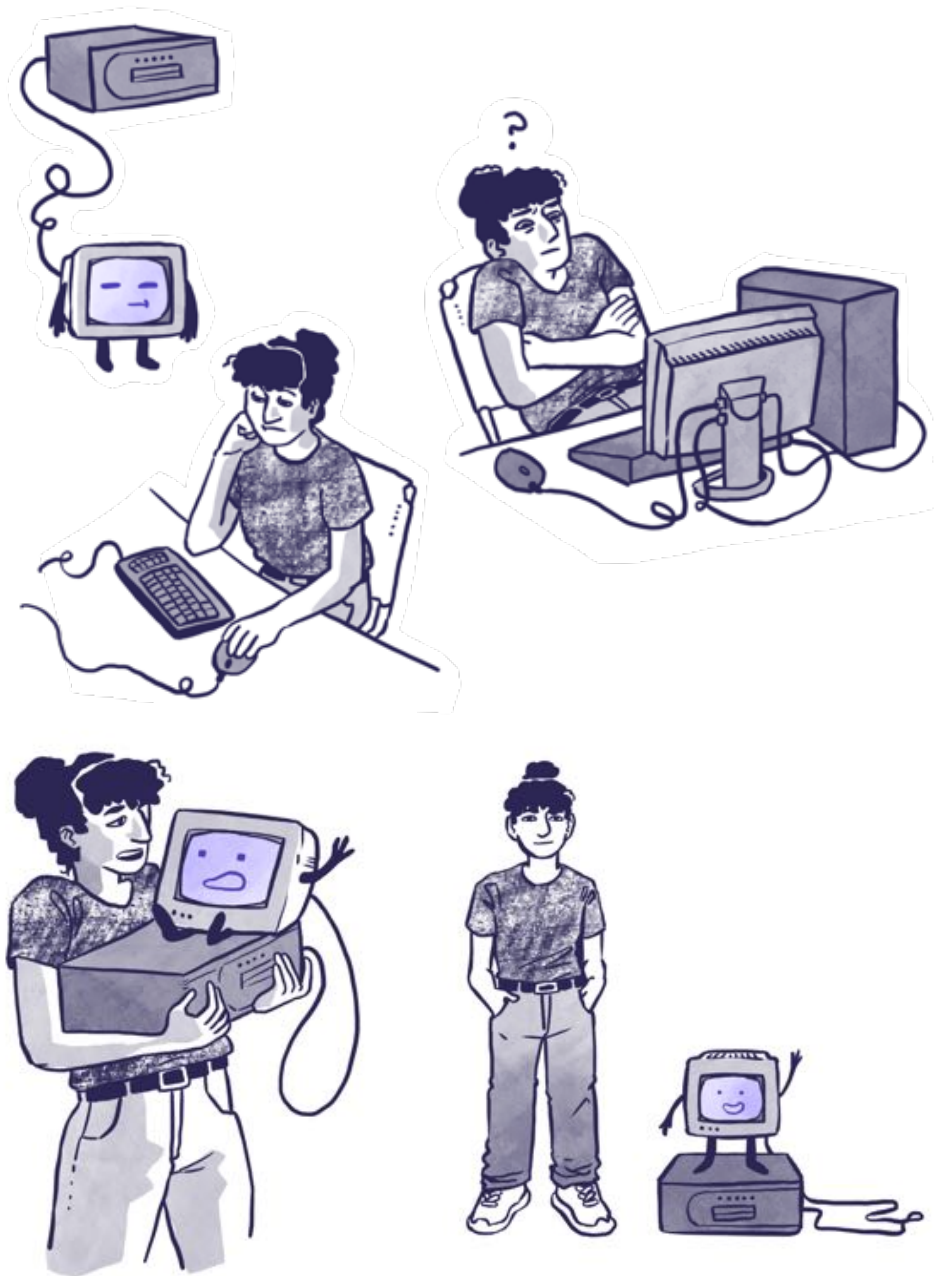
Figure 1: Drawing of the audience surrogate and computer

Preliminary sketches made when I was planning on having the comic be a conversation between an audience surrogate character—a confused student in a computer lab—and a sentient computer who answers her questions.

Another common framing device is to focus the comic on the life of someone relevant to a field of study. Through telling this person's story, the comic discusses their work and broader field. Generally, this figure (either historical or modern) is the narrator, and they explain their work to a fictional audience or directly to the reader. While there may be some kind of audience surrogate present, the notable figure(s) are the focus in this device, rather than the surrogate. Educational comics which employ this technique include *Logicomix* (Doxiadēs et al., 2009) and most of Jim Ottaviani's projects including *Feynman*, *Hawking*, *The Imitation Game: Alan Turing Decoded,* and *Primates: The Fearless Science of Jane Goodall, Dian Fossey, and Biruté Galdikas* each of which are focused on the eponymous scientist. This framing makes it possible for the author to provide history and broad context on the field. In *Welcome to Computer Science*, my focus was too broad to be able to select a single or a small number of historical figures to focus my comic on.

A third device which draws on aspects of the first two is to, rather than write about figures whose lives are inexorably tied to their scientific fields, write about fictional creatures who themselves are living examples of the topic. Through unfolding narratives about these fictional characters' lives, an author can fit in details about the scientific subject at hand. Examples of this technique are *Clan Apis* (2013)*,* about the lives of a hive of bees, and *The Last of the Sandwalkers* (2015)*,* about a beetle named Lucy going on an adventure, both by biologist and cartoonist Jay Hosler. Hosler uses his insect characters to tell emotionally rich stories about family, life, and exploration. They are worth reading as thoughtful fiction. Within these compelling narratives Hosler includes scientific facts about the insects' life cycles, patterns of behavior, and physical

form and a reader, regardless of their interest in the story's scientific content, comes away knowing more about insects. I didn't select this narrative frame for similar reasons to why I abandoned the audience surrogate frame; I didn't have a story I found compelling enough to commit to. This framing device also tends to deprioritize the science for the sake of a more compelling story, which didn't align with my goals for *Welcome to Computer Science.*

The last framing device I considered is to have a narrator directly explaining a topic to the reader. This narrator can either be a fictional character of some sort, the author themself, or a professional in the field that the author interviewed or worked with on the topic. Many comics use this method; notably, this is the preferred framing device of Larry Gonick, author of *The Cartoon History of the Universe* (1990) and nearly 20 other *Cartoon Guides* and *Histories.* Often Gonick's narrator is a wild-haired professor, though in some of his works the narrator represents his co-author, as in *Hypercapitalism: A Cartoon Critique of the Modern Economy and Its Values* (Gonick & Kasser, 2018). Scott McCloud's comic *Google Chrome* (2008) introducing Chrome and explaining how it differed from previous browsers, directly excerpts narration from interviews McCloud conducted with involved programmers and designers, producing a comic with many narrators.

Of course, many educational comics' framing devices fall into multiple of these categories—while *Logicomix* is centered on 20th century logician and philosopher Bertrand Russell narrating his life and work, there are segments where *Logicomix's* authors appear in the narrative and talk among themselves. Some comics do not use any of these techniques: Gonick's *The Cartoon Guide to Computer Science* (1983) does not

have a narrator at all, and instead is written in a style closer to an illustrated textbook. Not having a narrator at all would have made it difficult to create a comic that I was satisfied with. While I could have drawn images to go with text, there would have been few images that temporally connected to each other; it likely would have come out more like an illustrated textbook. Additionally, I like the conversational nature of having a narrator. I chose to go with what seemed to me the safest and most straightforward method, using a narrator without any other major characters, who is grounded in an identifiable physical space (a drawing of my own living room) but who also has the flexibility to appear in metaphorical panels, like a drawing of a compiler as a conveyor-belt machine.

Design

Narrator

Initially I thought the Narrator would be a computer monitor with little arms and legs coming directly out of it. However, after sketching the Narrator several times, as I figured out how to make them expressive, I drew a background and tried to put the Narrator into it. I realized that without a full body, the Narrator did not take up very much space. Giving the Narrator a humanoid body enabled me to use their body language as a communication tool.
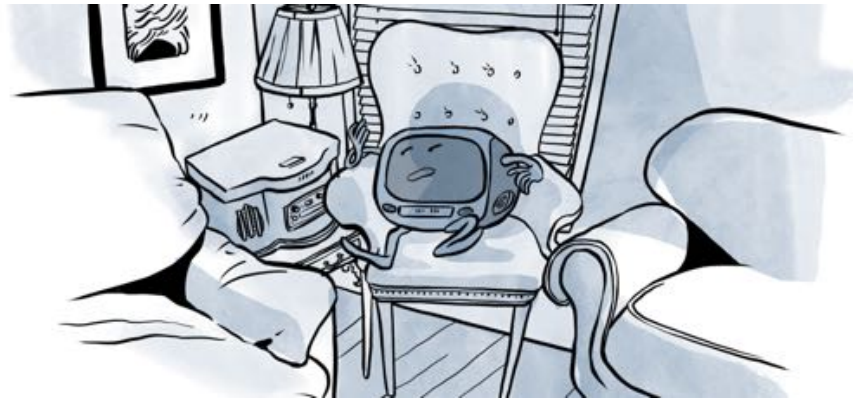
Figure 2: Sketching the Narrator

When drawing the Narrator in space, I realized their design would be more effective if they had a full body. They would be able to take up space in varied ways, and I could use their body language to convey emotion and information.

Environment

While working on the design of the computer, I was also developing the physical

location in which the comic would take place. I initially considered setting the comic in

a computer lab, because that seemed a natural location to learn about computer science.

However, a computer lab did not offer me many additional visual storytelling

opportunities. It was too on the nose. While sketching spaces I felt comfortable with, I

hit upon the living room environment. I like that the living room is comfortable and,

hopefully, inviting to the reader. I like the juxtaposition between the computer and the

homey environment. I based the living room off of the living room in my current home,

as a farewell to the space I have been inhabiting for so many pandemic months.

Visual Style of the Comic

I enjoy organic, imperfect lines. They make a comic look more tactile and

friendly. I also like the juxtaposition of the very concrete, artificial, exact nature of

computers and the very handmade look of imperfect drawings. Because of this, I knew I

wanted to draw the comic somewhat roughly, laying down the panel borders and word

balloons without straightedge tools. I was also invested in making the comic look hand

lettered. In the past I have gone through and lettered the whole comic myself, a time-

consuming process. Because *Welcome to Computer Science* is so long, I decided for

speed's sake I would make a font out of my handwriting, to get something that looked

more natural and handmade, but which would be significantly faster than lettering the

entire comic manually.

I chose to draw the comic primarily in black and white with relatively simple shading mostly for efficiency, but also because I feel a more subdued method of presentation (not highly rendered, not particularly colorful) helps keep the balance between words and images in a technical comic. It provides minimal distraction from the pertinent information.

Methods

To draw *Welcome to Computer Science* I used a Wacom 13HD Creative Pen & Touch Display (2015), with a large ergonomic grip on the tablet's pen. I used the software Clip Studio Paint EX because it supports multi-page documents and makes it simple to input and manipulate long scripts. I used the website Calligraphr to turn my handwriting into a font. Calligraphr supports having multiple alternate characters for each letter in the font, which makes the font look more naturally handwritten by ensuring that fewer of the letters are identical. However, Clip Studio Paint did not support alternate characters (or ligatures) in fonts, so I was not able to use this functionality, making the font I created less flexible than I hoped.

The comic is the industry standard graphic novel size at 6.625 inches by 10.25 inches, with a bleed width of 0.13 inches. The canvas has a DPI of 450. I inked the comic using the G-pen, set to a brush size of 27. The comic is lettered with a 12pt font.

I drew the comic with the guidance of a rough script, which in form was something between a paragraph description of each page and a film or theater script with dialogue and blocking description. In contrast to prose writing, which has approximately 250-350 words per page, a comic—even a very wordy one—has between 50 and 150 words per page.
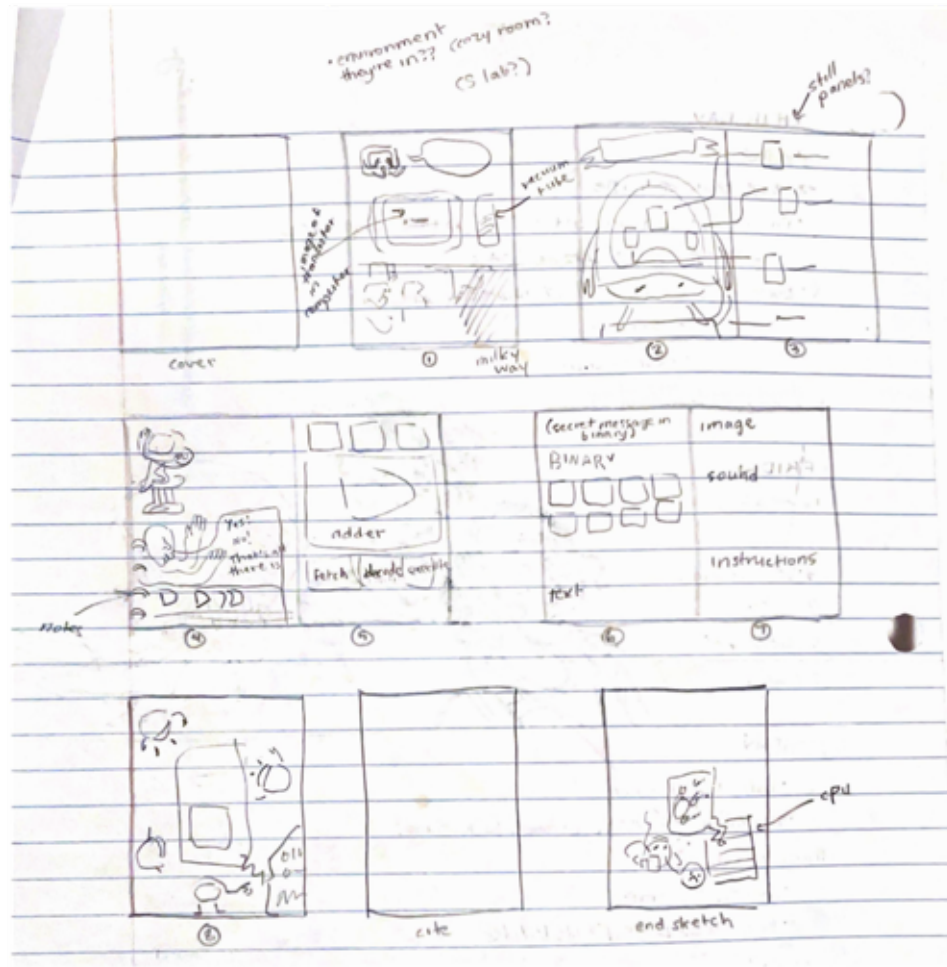
Figure 3: Part One thumbnails

The thumbnail sketch of the first section of *Welcome to Computer Science.* In the
completed comic many of the page layouts are different, but the topics each page is
focused on in the final comic is the same as it is presented here.

Thumbnailing

The thumbnailing stage of a comic refers to creating very small, quick sketches
of pages to figure out their layout without wasting time on refining drawings that would
be eventually discarded. After having an approximate script, I created thumbnails in
order to decide which portion of the script would appear on which page, and what
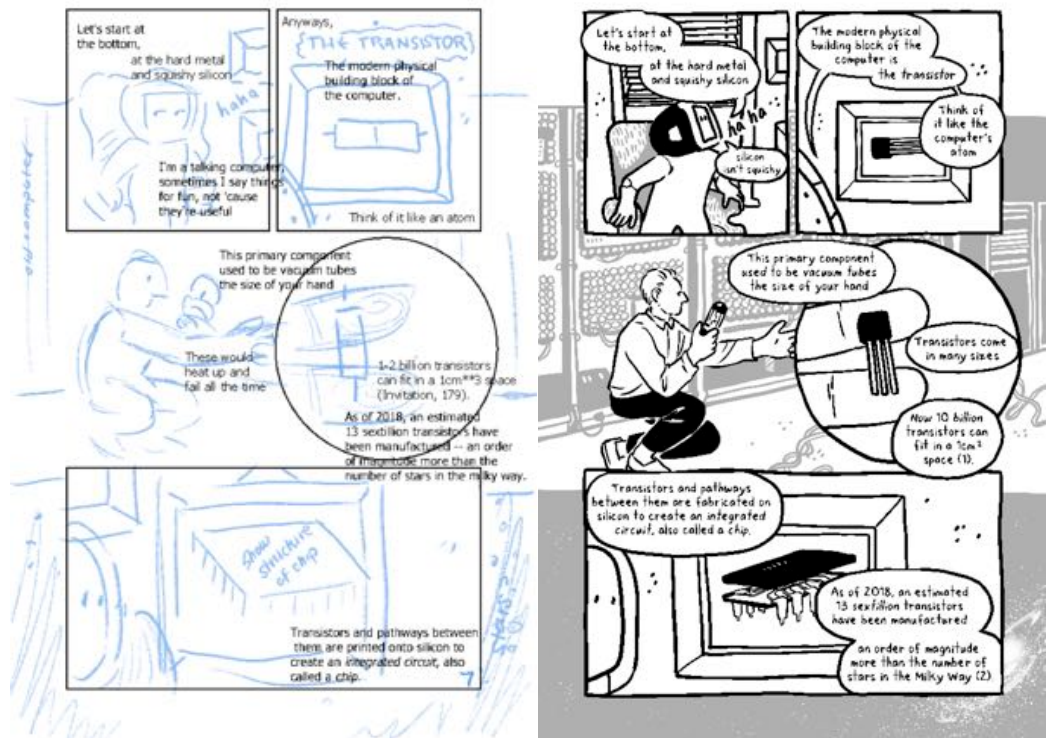aspects of the chapters I needed to cut.

Figure 4: Part One, page one during the sketch and inking stage

On the left is the preliminary rough pencil sketch. On the right is the version of the page after inking and adding many finishing details.

Penciling

The penciling stage can be broken into two parts. The first is to make a loose sketch based on the thumbnails, like the one seen above, on the left. This is when panel borders are created and the text from the script is roughly put into place. Then, for more complicated panels, I went back to the pencils and carefully sketched out the details of the image. This was particularly important for objects that required visual accuracy, like historical computers. Scenes that were mostly the Narrator in their apartment I tended to leave at the level of sketch seen above.

Inking

The inking stage consists of putting down darker lines and details. First, I inked the panel borders and word balloons so I could be certain of their placement. Then I drew the figures and backgrounds inside the panels. After I finished marking down the black lines of the comic, I went back through adding finishing details—mostly gray tones and other small pieces of visual interest. During this process I also edited the text to make it flow better.

Results

I created the 56-page comic *Welcome to Computer Science,* reproduced in full in the Appendix. The comic's cover shows a humanoid figure with a CRT monitor for a head (henceforth referred to as "the Narrator") holding open the door of a brick apartment building, welcoming the reader inside. There is a cat at the Narrator's feet.

After an introduction the body of the comic is broken into three parts, each part focused on a different topic: computer architecture, programming languages, and the internet. Each part has a title page, eight comic pages of material about the section, and a further reading page linking to external educational materials. This is followed by a conclusion.

Introduction

The introduction is three pages long, and introduces the Narrator, several elements of the Narrator's apartment which is the setting for most of the comic, and the comic's methodology: rather than introducing computers with "computational thinking," teaching the reader how to frame ideas and problems in terms of the logic structures computers use, this comic introduces the reader to the "breadth of computer science," explaining the machine from hardware to software.

Part One: What is a computer? No, literally, what is it?

Part one is about computer architecture. It introduces computer science by describing the computer itself, according to the "Von Neumann architecture," which defines the computer as being assembled from four essential parts. Then the comic

progresses to talking about how both logic and information (numbers, letters, images, and sound) are encoded using a binary representation, and finally it mentions how simple the commands that a computer "natively" understands are.

Part Two: There sure are a lot of programming languages, huh

Part two is about programming languages. Using illustrations of historical computers from the 19th century to the near-present, the comic discusses how programming languages and, more broadly, how interacting with computers through devices like keyboards and computer mice, has changed over time. Then it defines what machine code and assembly code are in relation to higher level programming languages and explains how compilers take higher-level languages and translate the code into something a particular computer can understand. The comic offers an explanation to why there are so many different programming languages by illustrating them on a map with a brief description of some of the benefits certain languages provide, and the problems the languages are best at solving. Lastly, the comic finally defines what programming language is in a roundabout manner, by sketching out a definition of "Turing completeness" and noting that if a system is not Turing complete, it is not a programming language.

Part Three: Everything is connected

Part three is about the internet. It covers the basic protocols underlying the internet which make it possible to reliably transfer information between machines and across long distances. Then it describes the infrastructure underlying the internet, that most users do not think about: the physical cables that bring Wi-Fi to internet users, and

the very concrete datacenters that store information that is typically thought of as in "the cloud." Next, it describes how the internet is made up of smaller networks, and how traffic is routed through this system. The section ends by differentiating the internet and the World Wide Web—the latter is built on top of the former.

Conclusion: And that's enough, for now

The conclusion to *Welcome to Computer Science* offers a look at the breadth of computer science by illustrating an (incomplete) map of different computer science sub-fields. Then, the comic ties together the sections by reiterating that the fundamental information computers manipulate is ones and zeros, and everything a computer does is built from these simple parts.

Future work

There is need for more studies focused on how comics can be best used in educational settings, and what kinds of educational comics are most useful for students. Particularly in computer science, studying what pedagogical tack is most useful for students, and how best to incorporate comics in a computer science course, would be useful in helping to design more effective and impactful educational comics.

I would like to expand *Welcome to Computer Science* to include more topics. I would like to complete the section I had planned on AI and Society, probably into more than eight pages and potentially breaking it into more manageable sub-topics. I would also like to create sections on other computer science topics. I could list any number of topics I would be interested in writing a section about, but foremost I am interested in cryptography, and talking about number theory and the ways math and computer

science intersect in this area. Many of the topics that I mentioned briefly in this comic, like compilers, could be expanded into sections of their own. I also want to make sections about the intersection of computers and society, including Net Neutrality, DRM, and the "hacker ethos," defining its historical meaning and looking at how that intersects with how it is understood in the broader culture. I could also focus in on much smaller topics that are often initially unintuitive to students, like recursion, where I could create a comic to help the reader visualize the process of recursion. With more sections added I could potentially publish this piece as a complete book, or, because of the work's modular nature, I could publish specific sections.

Contributions

I created the comic *Welcome to Computer Science,* which uses the framing device of a narrator talking to the reader to introduce computer science concepts chosen from the breadth of the field. Unlike other comics focused on computer science education, *Welcome to Computer Science* is a breadth-first introduction to computers and computing written for an undergraduate audience.

The comic does not expect previous knowledge of computer science. The comic introduces the reader to computer hardware, programming languages, and the internet, explaining these topics and visualizing the common metaphors used to explain them. This breadth-first approach provides the reader with a framework for understanding the field of computer science, and as such would be good preparatory reading before or concurrently with an introduction to programming class.

The comic strikes a balance between being accessible and communicating complex information, prioritizing clear explanation over technical terminology.

The study of the use of comics for teaching computer science is in its early stages. *Welcome to Computer Science* is a starting point for comic creators and educators looking for a method of creating comics about computer science, as well as researchers studying the impact of educational comics about computer science.

Appendix

Bibliography

Bach, B., Riche, N. H., Carpendale, S., & Pfister, H. (2017). The Emerging Genre of Data Comics. *IEEE Computer Graphics and Applications*, *37*(3), 6–13. https://doi.org/10.1109/MCG.2017.33

Chochois, H. (2021). *The body factory: From the first prosthetics to the augmented human*. The Pennsylvania State University Press, Graphic Mundi.

Collver, J., & Weitkamp, E. (2018). Alter egos: An exploration of the perspectives and identities of science comic creators. *Journal of Science Communication*, *17*(1), A01. https://doi.org/10.22323/2.17010201

Doxiadēs, A. K., Papadimitriou, C. H., Papadatos, A., & Di Donna, A. (2009). *Logicomix: An epic search for truth* (1st U.S. ed. 2009). Bloomsbury.

Evans, J. (n.d.). *Wizard zines*. Wizard Zines. Retrieved April 12, 2022, from https://wizardzines.com/

Farinella, M. (2018). The potential of comics in science communication. *Journal of Science Communication*, *17*(1), Y01. https://doi.org/10.22323/2.17010401

Farinella, M., & Roš, H. (2014). *Neurocomic* (First edition). Nobrow Press.

Gonick, L. (1983). *The cartoon guide to computer science*. Barnes & Noble.

Gonick, L., & Kasser, T. (2018). *Hypercapitalism: The modern economy, its values, and how to change them*. The New Press.

Griffith, S., Dragotta, N., Dragotta, I., Griffith, A., & Bonsen, J. (2014). *Howtoons: Tools of mass construction*. Image Comics.

Hosler, J. (2013). *Clan Apis*. Createspace.

Hosler, J. (2015). *Last of the sandwalkers*. First Second.

Justice, M. (2020). *How computers really work: A hands-on guide to the inner workings of the machine*. No Starch Press, Inc.

Kernighan, B. W. (2021). *Understanding the digital world: What you need to know about computers, the Internet, privacy, and security* (Second edition). Princeton University Press.

Liukas, L. (2015). *Hello Ruby: Adventures in coding*.

Marmion, J.-F., & Monsieur B. (2021). *Braincomix*. Graphic Mundi.

McCloud, S. (n.d.). *Google Chrome*. Retrieved April 12, 2022, from https://www.google.com/googlebooks/chrome/small_00.html

Ottaviani, J. (2011a). *Feynman*. First Second.

Ottaviani, J. (2011b, December 9). *Feynman—Science Study Break*. https://www.youtube.com/watch?v=s3OCm3GHUbI

Ottaviani, J., Myrick, L., & Polk, A. (2019). *Hawking*. First Second.

Ottaviani, J., & Purvis, L. (2016). *The imitation game: Alan Turing decoded*. Abrams ComicArts.

Ottaviani, J., & Wicks, M. (2013). *Primates: The fearless science of Jane Goodall, Dian Fossey, and Biruté Galdikas*. First Second.

Padua, S. (2015). *The thrilling adventures of Lovelace and Babbage: The (mostly) true story of the first computer*. Pantheon Books.

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M., & Paterson, J. (2007). A survey of literature on the teaching of introductory programming. *Working Group Reports on ITiCSE on Innovation and Technology in Computer Science Education*, 204–223. https://doi.org/10.1145/1345443.1345441

Schneider, G. M., & Gersting, J. L. (2019). *Invitation to computer science* (8th edition). Cengage Learning.

Singh, K., & Konak, I. (2018). *Ara the Star Engineer*.

Suh, S., Latulipe, C., Lee, K. J., Cheng, B., & Law, E. (2021). Using Comics to Introduce and Reinforce Programming Concepts in CS1. *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 369–375. https://doi.org/10.1145/3408877.3432465

Suh, S., Lee, M., Xia, G., & law, E. (2020). Coding Strip: A Pedagogical Tool for Teaching and Learning Programming Concepts through Comics. *2020 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 1–10. https://doi.org/10.1109/VL/HCC50065.2020.9127262

Vakil, S. (2018). Ethics, Identity, and Political Vision: Toward a Justice-Centered Approach to Equity in Computer Science Education. *Harvard Educational Review*, *88*(1), 26–52. https://doi.org/10.17763/1943-5045-88.1.26

White, N. (1999). *The Magic School Bus Gets Programmed.* Scholastic Inc.

Wibowo, A. (n.d.). *BubbleSort Zines*. BubbleSort Zines. Retrieved April 12, 2022, from https://shop.bubblesort.io/

Yang, G. L. (2018, June 15). *Comics belong in the classroom*. https://www.youtube.com/watch?v=xjvTIP7pV20

Yang, G. L., & Holmes, M. (2015). *Secret coders*. First Second.