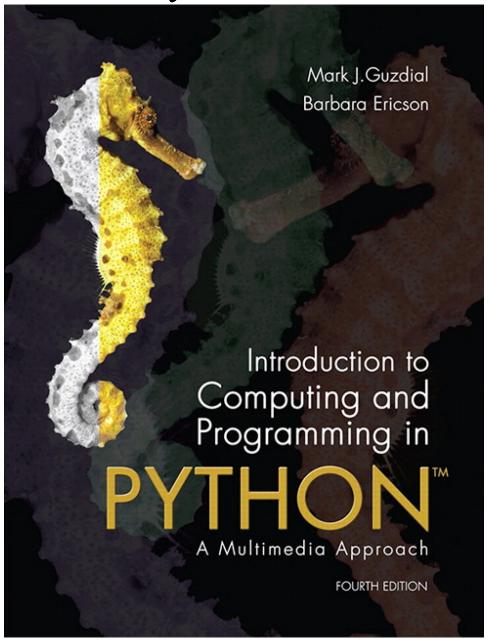
Solutions Manual for Introduction to Computin and Programming in Python: A Multimedia Approach, 4th Edition

By Mark J. Guzdial and Barbara Ericson



Solutions manual by Matthew Guzdial

Index

Chapter 1

1.1 *Every* profession uses computers today. Use a Web browser and a search engine like Google to find sites that relate your field of study with computer science or computing or computation. For example, search for "biology computer science" or "management computing."

A good answer should have at least 2-3 links. For example, for "biology" some good links would be:

http://www.brown.edu/research/projects/computational-molecular-biology/
http://www.sciencedirect.com/science/article/pii/S0960982201000811
http://www.biomedcentral.com/1471-2105/12/120

1.2 The 2013 Nobel Prize in Chemistry was in some sense a Nobel Prize given for work in computer science. What was the role of computers in that Nobel Prize?

Computers were used to simulate the behavior of large molecules in a brand new way combining classical and quantum physics.

(Source: http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/press.html)

1.3 Text characters are encoded in different ways. The bottom level is always binary in bytes, but a different binary pattern can be used to represent different characters. Two of these encodings are *ASCII* and *Unicode*. See if you can do Web searches on ASCII and Unicode. What's the difference between ASCII and Unicode? Why would we need Unicode if we already had ASCII?

The major difference between ASCII and Unicode is the amount of characters both hold. ASCII has 128 characters, while Unicode has over 110,000 characters. Unicode is largely used for non-Latin letters, such as Chinese characters,

(Sources: http://en.wikipedia.org/wiki/ASCII and http://en.wikipedia.org/wiki/Unicode)

1.4 Consider the representation for pictures described in Section 1.4, where each dot (pixel) in the picture is represented by three bytes, for the red, green, and blue components of the color at that dot. How many bytes does it take to represent a 640 by 480 picture, a common picture size on the Web? How many bytes does it take to represent a 1024 by 768 pictures, a common screen size? (What do you think is meant now by a "three megapixel" camera?)

Bytes for 640 by 480 picture: (640*480)*3 = 921,600
Bytes for a 1024 by 768: (1024*768)*3 = 2,359,296

A "three megapixel" camera means that the camera can capture images with three million pixels.

1.5 One bit can represent 0 or 1. With two bits you have four possible combinations 00, 01, 10, and 11. How many different combinations can you make with four bits or eight bits (one byte)? Each combination can be used to represent a binary number. How many numbers can you represent with 2 bytes (16 bits)? How many numbers can you represent with four bytes?

Four bits: 16 combinations (0000, 0001, 0010, 0100, 1000, 0011, 0110, 1100, 0101, 1010, 1001, 0111, 1110, 1101, 1011, 1111)
Eight bits: 256 combinations
Two bytes/16 bits: 65,536 combinations
Four bytes/32 bits: 4,294,967,296 combinations

1.6 Microsoft Word *used* to use an encoding of a word-processing document called "DOC." Most recent versions of Microsoft Word use a different encoding called "DOCX." What's the difference between them?

The "DOC" encoding was a unique encoding to Microsoft, while the "DOCX" encoding is based on XML, the Extendable Markup Language. This makes it easier for other programs to parse "DOCX" documents, along with conferring a number of other benefits (such as smaller size, higher quality images, and a lower chance of file corruption).

(Source: http://www.inc.com/software/articles/201002/doc.html and http://www.thebookdesigner.com/2013/04/docx-vs-doc/)

1.7 One of the powerful ideas in computer science is that encodings can be *layered*. Most of the encodings we've talked about so far (e.g., pixels in a picture, characters, floating point numbers) are based on binary. *XML* is a way of encoding information in text, which is in turn encoded in binary. What are the advantages and disadvantages of using XML rather than binary encodings?

Advantages:
-XML is easy for humans and computers to read.
-XML is easily extendible
-XML has been rapidly adopted by industry
Disadvantages:
-XML takes up more space than binary encoding
-XML is extendible and therefore not as standardized

(Source: http://www.mulberrytech.com/papers/HowAndWhyXML/slide009.html)

1.8 How can you encode a *floating-point number* in terms of bytes? Do a search on the Web for "floating point." You will find that there are different encodings of floating point numbers. Take a common one like the IEEE Floating Point Standard as an example. Assuming *single precision*, what is the largest and smallest numbers that you can represent in that encoding?
A computer can represent a floating-point number as it would an integer number, but with some extra information. Specifically, to be able to define the decimal portions of a floating point, a computer must know an extra integer to raise 10 by to multiply the base integer by. For example, if one was to represent the number 0.5 it would hold track of the number 5 and -1, since 0.5 equals 5 x10^(-1).

Smallest number: $1\text{x}10^{-101}$
Largest number: $9999999\text{x}10^{90}$

(Source: http://www.wikipedia.org/wiki/IEEE_floating_point)

1.9 As we said in the chapter, computer science is about *people*. Start your exploration of computer science by exploring the people who are computer scientists and influence computer science. As you do, find Web sites that you believe are *credible* for your information, and include the URL in your answers—with your reasons for believing that source is credible. What general rules do you use to determine what is a credible Web site?

Note: This question can be potentially confusing. Due to its structure, multiple interpretations are possible. There are essentially two sections, the first implicitly asking students to find people involved in computer science (and include the credible URL where they found them). The second part asks how to determine if a Web site is credible.

Some examples of people involved in computer science (and links):
-Mark Guzdial (http://www.cc.gatech.edu/fac/mark.guzdial/)
-Anita Borg (http://anitaborg.org/)
-Bill Gates, Mark Zuckerberg, and Susan Wojcicki (http://www.mercurynews.com/business/ci_24308090/code-org-mark-zuckerberg-bill-gates-teach-computer-science-every-school)

Ways to determine if a link is credible:
-If it ends with a ".edu"
-If the link is a primary website for an organization or individual
-If the link references other credible web sites

1.10 Look up Alan Kay, *object-oriented programming*, and the *Dynabook* on the Web. Alan is one of the inspirations for the media computation approach that we use in this book. Can you figure out what he has to do with media computation?

There are several potential answers to this question:
-Alan Kay is considered one of the fathers of object oriented programming, which is used exclusively in this book (Source: http://amturing.acm.org/award_winners/kay_3972189.cfm)
-Alan Kay is invested in educating all people in computer science (Source: http://www.vpri.org/)
-Perhaps the best answer, Alan Kay is interested in making use of media-rich environments to teach computer science (Source: http://www.squeakland.org/)

1.11 Look up Clarence (Skip) Ellis. Without him, Google Docs wouldn't work the way that they do to help people collaborate. What did he do?

Clarence Ellis is considered one of the pioneers of operational transformation, which deals with collaborative systems, such as those now used in Google Docs to allow for multi-user editing. (Source: https://cs.illinois.edu/news/memory-clarence-ellis-1943-2014)

1.12 Look up Grace Hopper on the Web. How did she contribute to programming languages?

Grace Hopper created the first compiler for a computer programming language and coined the term "debugging" for fixing computer flitches. (Source: http://en.wikipedia.org/wiki/Grace_Hopper#cite_note-Booss03-6)

1.13 Look up Andrea Lawrence on the Web. What computer science department did she chair?

Andrea Lawrence chaired the computer and information sciences department of Spelman University (Source: http://www.thehistorymakers.com/biography/andrea-lawrence-42)

1.14 Look up Alan Turing on the Web. What does he have to do with our notion of what a computer can do and how encodings work?

Alan Turing created the definition for a modern computer in terms of what it can do, referred to as a "Turing Machine". Additionally, he came up with a means of encoding an action table of a Turing machine as a string, such that someone can feed in instructions to a machine and get an output back. (Source: http://en.wikipedia.org/wiki/Universal_Turing_machine)

1.15 Look up Adele Goldberg on the Web. How did she contribute to programming languages?

She was a forerunner or pioneer in a number of now common facets of programming languages, including design patterns, object-oriented programming, and graphical user interfaces. (Source: http://en.wikipedia.org/wiki/Adele_Goldberg_%28computer_scientist%29)

1.16 Look up Kurt Gödel on the Web. What amazing things did he do with encodings?

He created Gödel numbering, a means by which a sequence of symbols can be represented as natural numbers (Source: http://en.wikipedia.org/wiki/G%C3%B6del_numbering)

1.17 Look up Ada Lovelace on the Web. What amazing things did she do before the first mechanical computer was built?

Ada Lovelace was the first ever computer programmer, before the first computer was even built, having come up with a stepwise sequence of operations for solving mathematical problems. (Source: http://www.computerhistory.org/babbage/adalovelace/)

1.18 Look up Claude Shannon on the Web. What did he do for his master's thesis?

Claude Shannon proved in his master's thesis that Boolean algebra could be used to simplify relays used in the electromechanical phone exchanges of the day, and later defined digital circuit design. (Source: http://dspace.mit.edu/bitstream/handle/1721.1/11173/34541425.pdf?sequence=1)

1.19 Look up Richard Tapia on the Web. What has he done to encourage diversity in computing?

Richard Tapia directed or co-directed more minority and women doctoral candidates in science and engineering than anyone in the country and has lead numerous programs to increase diversity in the fields of science and engineering. (Source: http://tapiaconference.org/about/tapia)

1.20 Look up Marissa Mayer on the Web. What computer tool (that you probably use regularly) did she help create?

She helped to create Google's now famous unadorned search page, along with the look and feel of some of the company's most popular products. (Source: http://dealbook.nytimes.com/2012/07/16/googles-marissa-mayer-tapped-as-yahoos-chief/?hp)

1.21 Look up Shafi Goldwasser on the Web. What major computing prize did she win in 2012 and why?

The ACM Turing Award, which she won for work in the theoretic foundations for the science cryptography and pioneered new ways to check mathematical proofs. (Source: http://amturing.acm.org/award_winners/goldwasser_8627889.cfm)

1.22 Look up Mary Lou Jepsen on the Web. What new technology is she working on?

She is working on small screens that can seamlessly join together like legos to create larger screens, along with wall screens. (Source: http://www.fastcodesign.com/3036692/fast-feed/google-is-inventing-screens-that-snap-together-like-lego)

1.23 Look up Ashley Qualls on the Web. What did she create that is worth a million dollars?

She created a website as a teenager that offered free MySpace account designs, making thousands of dollars from ad revenue alone. (Source: https://web.archive.org/web/20080220145135/http://potw.news.yahoo.com/s/potw/52250/teen-millionaire)

1.24 Look up Tim Berners-Lee on the Web. What did he invent?

He implemented a communication protocol (Hypertext Transfer Protocol [HTTP]) that allowed for client and server communication via the Internet, thereby inventing the World Wide Web. (Source: http://www.w3.org/People/Berners-Lee/)

1.25 As in every field, people in computer science build on one another's work.

. Who invented the Logo Turtle?

  Paul Wexelblat built the first turtle (Source: http://web.archive.org/web/20090310020335/http://www.erzwiss.uni-hamburg.de/Sonstiges/Logo/logofaqx.htm#FAQ2)

. Who used the Logo Turtle to have fourth graders learn about fractions in mathematics?

There are multiple answers to this question, including Idit Harel, or Yasmin B. Kafai. (Source: http://investigations.terc.edu/library/bookpapers/research_on_logo.cfm)

. Who invented a programming language that featured thousands of Logo Turtles in order to model complex behavior like ants and termites?

Uri Wilensky authored NetLogo, which allows for simulation and modeling of ant behavior and more. (Source: http://ccl.northwestern.edu/netlogo/index.shtml)

1.26 Now trace this series of who built on whose work.

. Who invented the laser printer?

Gary Starkweather invented the laser printer. (Source: http://xenia.media.mit.edu/~yarin/laser/laser_printing.html)

. One of the winners of the ACM Turing Award (the closest that computer science has to a Nobel Prize) invented a computer system for typesetting books on a laser printer. Who was that?

Donald Knuth built a system (referred to as TeX) to computationally typeset books for a laser printer.
(Source: http://amturing.acm.org/award_winners/knuth_1013846.cfm)

. The winner of a recent ACM Turing Award built a computer system *on top of* the last typesetting system, to make it easier to use (but that's not what he won the Turing Award for). Who was that, and what did he win his award for?

Leslie Lamport developed the LaTeX system written on top of TeX. He won his Turing Award in 2013 for contributions to distributed and concurrent systems, essentially when several computations are executed simultaneously, potentially interacting.
(Source: http://amturing.acm.org/award_winners/lamport_1205376.cfm)

Chapter 2

2.1

What is an algorithm?

Algorithm: A description of a process in a step-by-step manner, not tied to any programming language. The same algorithm may be implemented in many different languages in many different ways in many different programs—but they would all be the same process if we're talking about the same algorithm. (Chapter 1, Page 9)