Curriculum (/bjc-course/curriculum) / Unit 1 (/bjc-course/curriculum/01-welcome) /

Unit 1: Welcome to Visual Programming

Broadcast, Animations, and Music!

Learning Objectives

- 1: The student can use computing tools and techniques to create artifacts.
- 4: The student can use programming as a creative tool.
- 5: The student can describe the combination of abstractions used to represent data.
- 6: The student can explain how binary sequences are used to represent digital data.
- 7: The student can develop an abstraction.
- 9: The student can use models and simulations to raise and answer questions.
- 28: The student can analyze how computing affects communication, interaction, and cognition.
- 29: The student can connect computing with innovations in other fields.
- 30: The student can analyze the beneficial and harmful effects of computing.
- 31: The student can connect computing within economic, social, and cultural contexts.

Readings/Lectures

- Blown to Bits: Chapter 1 (http://www.bitsbook.com/wp-content/uploads/2008/12/chapter1.pdf)
- Reading 1.01: What is Abstraction? (/bjc-course/curriculum/01-welcome/readings/01-what-is-abstraction)
- Reading 1.02: More on Abstraction (/bjc-course/curriculum/01-welcome/readings/02-more-on-abstraction)
- Reading 1.03: Binary and Hexadecimal Numbers (/bjc-course/curriculum/01-welcome/readings/03-binary-and-hexadecimal)

External Resources

- Lecture Video: Binary Hex Decimal (http://www.screencast.com/t/c2tp610y1tx6)
- Why software is eating the world (http://online.wsj.com/article/SB10001424053111903480904576512250915629460.html) (Wall Street Journal) (DEAD LINK)

Labs/Exercises

- Lab 1.01: Conversion Exercise (/bjc-course/curriculum/01-welcome/labs/01-conversion)
- Lab 1.02: Welcome to Visual Programming (/bjc-course/curriculum/01-welcome/labs/02-exploring-visual-programming)
- Lab 1.03: Lights, Camera, Action (/bjc-course/curriculum/01-welcome/labs/03-lights-camera-action)

Blown to Bits

Your Life, Liberty, and Happiness After the Digital Explosion

Hal Abelson
Ken Ledeen
Harry Lewis

Addison-Wesley

Upper Saddle River, NJ • Boston • Indianapolis • San Francisco
New York • Toronto • Montreal • London • Munich • Paris • Madrid
Cape Town • Sydney • Tokyo • Singapore • Mexico City

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside the United States, please contact:

International Sales
international@pearson.com

Visit us on the Web: www.informit.com/aw

Library of Congress Cataloging-in-Publication Data:

Abelson, Harold.
Blown to bits : your life, liberty, and happiness after the digital explosion / Hal Abelson, Ken Ledeen, Harry Lewis.
p. cm.
ISBN 0-13-713559-9 (hardback : alk. paper) 1. Computers and civilization. 2. Information technology—Technological innovations. 3. Digital media. I. Ledeen, Ken, 1946- II. Lewis, Harry R. III. Title.
QA76.9.C66A245 2008
303.48'33—dc22

2008005910

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax (617) 671 3447

ISBN-13: 978-0-13-713559-2
ISBN-10: 0-13-713559-9
Text printed in the United States on recycled paper at RR Donnelley in Crawfordsville, Indiana.
Third printing December 2008

This Book Is Safari Enabled

The Safari® Enabled icon on the cover of your favorite technology book means the book is available through Safari Bookshelf. When you buy this book, you get free access to the online edition for 45 days.

Safari Bookshelf is an electronic reference library that lets you easily search thousands of technical books, find code samples, download chapters, and access technical information whenever and wherever you need it.

To gain 45-day Safari Enabled access to this book:

- Go to http://www.informit.com/onlineedition
- Complete the brief registration form
- Enter the coupon code 9SD6-IQLD-ZDNI-AGEC-AG6L

If you have difficulty registering on Safari Bookshelf or accessing the online edition, please e-mail customer-service@safaribooksonline.com.

Editor in Chief
Mark Taub

Acquisitions Editor
Greg Doench

Development Editor
Michael Thurston

Managing Editor
Gina Kanouse

Senior Project Editor
Kristy Hart

Copy Editor
Water Crest Publishing, Inc.

Indexer
Erika Millen

Proofreader
Williams Woods Publishing Services

Publishing Coordinator
Michelle Housley

Interior Designer and Composition
Nonie Ratcliff

Cover Designer
Chuti Prasertsith

CHAPTER 1

Digital Explosion

Why Is It Happening, and What Is at Stake?

On September 19, 2007, while driving alone near Seattle on her way to work, Tanya Rider went off the road and crashed into a ravine.* For eight days, she was trapped upside down in the wreckage of her car. Severely dehydrated and suffering from injuries to her leg and shoulder, she nearly died of kidney failure. Fortunately, rescuers ultimately found her. She spent months recuperating in a medical facility. Happily, she was able to go home for Christmas.

Tanya's story is not just about a woman, an accident, and a rescue. It is a story about bits—the zeroes and ones that make up all our cell phone conversations, bank records, and everything else that gets communicated or stored using modern electronics.

Tanya was found because cell phone companies keep records of cell phone locations. When you carry your cell phone, it regularly sends out a digital "ping," a few bits conveying a "Here I am!" message. Your phone keeps "pinging" as long as it remains turned on. Nearby cell phone towers pick up the pings and send them on to your cellular service provider. Your cell phone company uses the pings to direct your incoming calls to the right cell phone towers. Tanya's cell phone company, Verizon, still had a record of the last location of her cell phone, even after the phone had gone dead. That is how the police found her.

So why did it take more than a week?

If a woman disappears, her husband can't just make the police find her by tracing her cell phone records. She has a privacy right, and maybe she has good reason to leave town without telling her husband where she is going. In

* Citations of facts and sources appear at the end of the book. A page number and a phrase identify the passage.

Tanya's case, her bank account showed some activity (more bits!) after her disappearance, and the police could not classify her as a "missing person." In fact, that activity was by her husband. Through some misunderstanding, the police thought he did not have access to the account. Only when the police suspected Tanya's husband of involvement in her disappearance did they have legal access to the cell phone records. Had they continued to act on the true presumption that he was blameless, Tanya might never have been found.

New technologies interacted in an odd way with evolving standards of privacy, telecommunications, and criminal law. The explosive combination almost cost Tanya Rider her life. Her story is dramatic, but every day we encounter unexpected consequences of data flows that could not have happened a few years ago.

When you have finished reading this book, you should see the world in a different way. You should hear a story from a friend or on a newscast and say to yourself, "that's really a bits story," even if no one mentions anything digital. The movements of physical objects and the actions of flesh and blood human beings are only the surface. To understand what is really going on, you have to see the virtual world, the eerie flow of bits steering the events of life.

This book is your guide to this new world.

The Explosion of Bits, and Everything Else

The world changed very suddenly. Almost everything is stored in a computer somewhere. Court records, grocery purchases, precious family photos, pointless radio programs.... Computers contain a lot of stuff that isn't useful today but somebody thinks might someday come in handy. It is all being reduced to zeroes and ones—"bits." The bits are stashed on disks of home computers and in the data centers of big corporations and government agencies. The disks can hold so many bits that there is no need to pick and choose what gets remembered.

So much digital information, misinformation, data, and garbage is being squirreled away that most of it will be seen only by computers, never by human eyes. And computers are getting better and better at extracting meaning from all those bits—finding patterns that sometimes solve crimes and make useful suggestions, and sometimes reveal things about us we did not expect others to know.

The March 2008 resignation of Eliot Spitzer as Governor of New York is a bits story as well as a prostitution story. Under anti-money laundering (AML) rules, banks must report transactions of more than $10,000 to federal regulators. None of Spitzer's alleged payments reached that threshold, but his bank's computer found that transfers of smaller sums formed a suspicious pattern. The AML rules exist to fight terrorism and organized crime. But while the computer was monitoring small banking transactions in search of big-time crimes, it exposed a simple payment for services rendered that brought down the Governor.

Once something is on a computer, it can replicate and move around the world in a heartbeat. Making a million perfect copies takes but an instant—copies of things we want everyone in the world to see, and also copies of things that weren't meant to be copied at all.

The digital explosion is changing the world as much as printing once did—and some of the changes are catching us unaware, blowing to bits our assumptions about the way the world works.

When we observe the digital explosion at all, it can seem benign, amusing, or even utopian. Instead of sending prints through the mail to Grandma, we put pictures of our children on a photo album web site such as Flickr. Then not only can Grandma see them—so can Grandma's friends and anyone else. So what? They are cute and harmless. But suppose a tourist takes a vacation snapshot and you just happen to appear in the background, at a restaurant where no one knew you were dining. If the tourist uploads his photo, the whole world could know where you were, and when you were there.

Data leaks. Credit card records are supposed to stay locked up in a data warehouse, but escape into the hands of identity thieves. And we sometimes give information away just because we get something back for doing so. A company will give you free phone calls to anywhere in the world—if you don't mind watching ads for the products its computers hear you talking about.

And those are merely things that are happening today. The explosion, and the social disruption it will create, have barely begun.

We already live in a world in which there is enough memory *just in digital cameras* to store every word of every book in the Library of Congress a hundred times over. So much email is being sent that it could transmit the full text of the Library of Congress in ten minutes. Digitized pictures and sounds take more space than words, so emailing all the images, movies, and sounds might take a year—but that is just today. The explosive growth is still happening. Every year we can store more information, move it more quickly, and do far more ingenious things with it than we could the year before.

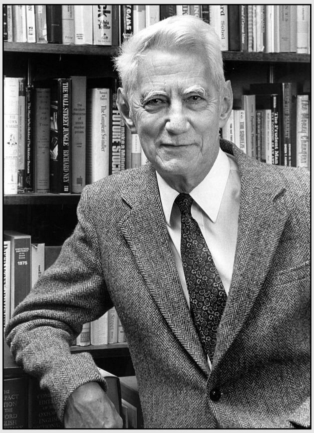
So much disk storage is being produced every year that it could be used to record a page of information, every minute or two, about you *and every other human being on earth*. A remark made long ago can come back to haunt a political candidate, and a letter jotted quickly can be a key discovery for a biographer. Imagine what it would mean to record every word every human being speaks or writes in a lifetime. The technological barrier to that has already been removed: There is enough storage to remember it all. Should any social barrier stand in the way?

Sometimes things seem to work both better and worse than they used to. A "public record" is now *very* public—before you get hired in Nashville, Tennessee, your employer can figure out if you were caught ten years ago taking an illegal left turn in Lubbock, Texas. The old notion of a "sealed court record" is mostly a fantasy in a world where any tidbit of information is duplicated, cataloged, and moved around endlessly. With hundreds of TV and radio stations and millions of web sites, Americans love the variety of news sources, but are still adjusting uncomfortably to the displacement of more authoritative sources. In China, the situation is reversed: The technology creates greater government control of the information its citizens receive, and better tools for monitoring their behavior.

This book is about how the digital explosion is changing everything. It explains the technology itself—why it creates so many surprises and why things often don't work the way we expect them to. It is also about things the information explosion is destroying: old assumptions about our privacy, about our identity, and about who is in control of our lives. It's about how we got this way, what we are losing, and what remains that society still has a chance to put right. The digital explosion is creating both opportunities and risks. Many of both will be gone in a decade, settled one way or another. Governments, corporations, and other authorities are taking advantage of the chaos, and most of us don't even see it happening. Yet we all have a stake in the outcome. Beyond the science, the history, the law, and the politics, this book is a wake-up call. The forces shaping your future are digital, and you need to understand them.

The Koans of Bits

Bits behave strangely. They travel almost instantaneously, and they take almost no space to store. We have to use physical metaphors to make them understandable. We liken them to dynamite exploding or water flowing. We even use social metaphors for bits. We talk about two computers agreeing on some bits, and about people using burglary tools to steal bits. Getting the right metaphor is important, but so is knowing the limitations of our metaphors. An imperfect metaphor can mislead as much as an apt metaphor can illuminate.



CLAUDE SHANNON

Claude Shannon (1916–2001) is the undisputed founding figure of information and communication theory. While working at Bell Telephone Laboratories after the Second World War, he wrote the seminal paper, "A mathematical theory of communication," which foreshadowed much of the subsequent development of digital technologies. Published in 1948, this paper gave birth to the now-universal realization that the bit is the natural unit of information, and to the use of the term.

Alcatel-Lucent, `http:www.bell-labs.com/news/2001/february/26/shannon2_lg.jpeg`.

We offer seven truths about bits. We call them "koans" because they are paradoxes, like the Zen verbal puzzles that provoke meditation and enlightenment. These koans are oversimplifications and over-generalizations. They describe a world that is developing but hasn't yet fully emerged. But even today they are truer than we often realize. These themes will echo through our tales of the digital explosion.

Koan 1: It's All Just Bits

Your computer successfully creates the illusion that it contains photographs, letters, songs, and movies. All it really contains is bits, lots of them, patterned in ways you can't see. Your computer was designed to store just bits—all the files and folders and different kinds of data are illusions created by computer programmers. When you send an email containing a photograph, the computers that handle your message as it flows through the Internet have no idea that what they are handling is part text and part graphic. Telephone calls are also just bits, and that has helped create competition—traditional phone companies, cell phone companies, cable TV companies, and Voice over IP (VoIP) service providers can just shuffle bits around to each other to complete calls. The Internet was designed to handle just bits, not emails or attachments, which are inventions of software engineers. We couldn't live without those more intuitive concepts, but they are artifices. Underneath, it's all just bits.

This koan is more consequential than you might think. Consider the story of Naral Pro-Choice America and Verizon Wireless. Naral wanted to form a text messaging group to send alerts to its members. Verizon decided not to allow it, citing the "controversial or unsavory" things the messages might contain. Text message alert groups for political candidates it would allow, but not for political causes it deemed controversial. Had Naral simply wanted telephone service or an 800 number, Verizon would have had no choice. Telephone companies were long ago declared "common carriers." Like railroads, phone companies are legally prohibited from picking and choosing customers from among those who want their services. In the bits world, there is no difference between a text message and a wireless phone call. It's all just bits, traveling through the air by radio waves. But the law hasn't caught up to the technology. It doesn't treat all bits the same, and the common carriage rules for voice bits don't apply to text message bits.

Verizon backed down in the case of Naral, but not on the principle. A phone company can do whatever it thinks will maximize its profits in deciding whose messages to distribute. Yet no sensible engineering distinction can be drawn between text messages, phone calls, and any other bits traveling through the digital airwaves.

EXCLUSIVE AND RIVALROUS

Economists would say that bits, unless controlled somehow, tend to be *non-exclusive* (once a few people have them, it is hard to keep them from others) and *non-rivalrous* (when someone gets them from me, I don't have any less). In a letter he wrote about the nature of ideas, Thomas Jefferson eloquently stated both properties. *If nature has made any one thing less susceptible than all others of exclusive property, it is the action of the thinking power called an idea, which an individual may exclusively possess as long as he keeps it to himself; but the moment it is divulged, it forces itself into the possession of every one, and the receiver cannot dispossess himself of it. Its peculiar character, too, is that no one possesses the less, because every other possesses the whole of it.*

Koan 2: Perfection Is Normal

To err is human. When books were laboriously transcribed by hand, in ancient scriptoria and medieval monasteries, errors crept in with every copy. Computers and networks work differently. Every copy is perfect. If you email a photograph to a friend, the friend won't receive a fuzzier version than the original. The copy will be identical, down to the level of details too small for the eye to see.

Computers do fail, of course. Networks break down too. If the power goes out, nothing works at all. So the statement that copies are normally perfect is only relatively true. Digital copies are perfect only to the extent that they can be communicated at all. And yes, it is possible in theory that a single bit of a big message will arrive incorrectly. But networks don't just pass bits from one place to another. They check to see if the bits seem to have been damaged in transit, and correct them or retransmit them if they seem incorrect. As a result of these error detection and correction mechanisms, the odds of an actual error—a character being wrong in an email, for example—are so low that we would be wiser to worry instead about a meteor hitting our computer, improbable though precision meteor strikes may be.

The phenomenon of perfect copies has drastically changed the law, a story told in Chapter 6, "Balance Toppled." In the days when music was distributed on audio tape, teenagers were not prosecuted for making copies of songs, because the copies weren't as good as the originals, and copies of copies would be even worse. The reason that thousands of people are today receiving threats from the music and movie industries is that their copies are perfect—not just as good as the original, but identical to the original, so that even the notion of "original" is meaningless. The dislocations caused by file sharing are not over yet. The buzzword of the day is "intellectual property." But bits are an odd kind of property. Once I release them, everybody has them. And if I give you my bits, I don't have any fewer.

Koan 3: There Is Want in the Midst of Plenty

Vast as world-wide data storage is today, five years from now it will be ten times as large. Yet the information explosion means, paradoxically, the loss of information that is not online. One of us recently saw a new doctor at a clinic he had been using for decades. She showed him dense charts of his blood chemistry, data transferred from his home medical device to the clinic's computer—more data than any specialist could have had at her disposal five years ago. The doctor then asked whether he had ever had a stress test and what the test had shown. Those records should be all there, the patient explained, in the medical file. But it was in the *paper* file, to which the doctor did not have access. It wasn't in the *computer's* memory, and the patient's memory was being used as a poor substitute. The old data might as well not have existed at all, since it wasn't digital.

Even information that exists in digital form is useless if there are no devices to read it. The rapid progress of storage engineering has meant that data stored on obsolete devices effectively ceases to exist. In Chapter 3, "Ghosts in the Machine," we shall see how a twentieth-century update of the eleventh-century British Domesday Book was useless by the time it was only a sixtieth the age of the original.

Or consider search, the subject of Chapter 4, "Needles in the Haystack." At first, search engines such as Google and Yahoo! were interesting conveniences, which a few people used for special purposes. The growth of the World Wide Web has put so much information online that search engines are for many people the first place to look for something, before they look in books or ask friends. In the process, appearing prominently in search results has become a matter of life or death for businesses. We may move on to purchase from a competitor if we can't find the site we wanted in the first page or two of results. We may assume something didn't happen if we can't find it quickly in an online news source. If it can't be found—and found quickly—it's just as though it doesn't exist at all.

Koan 4: Processing Is Power

The speed of a computer is usually measured by the number of basic operations, such as additions, that can be performed in one second. The fastest computers available in the early 1940s could perform about five operations per second. The fastest today can perform about a trillion. Buyers of personal computers know that a machine that seems fast today will seem slow in a year or two.

For at least three decades, the increase in processor speeds was exponential. Computers became twice as fast every couple of years. These increases were one consequence of "Moore's Law" (see sidebar).

Since 2001, processor speed has not followed Moore's Law; in fact, processors have hardly grown faster at all. But that doesn't mean that computers won't continue to get faster. New chip designs include multiple processors on the same chip so the work can be split up and performed in parallel. Such design innovations promise to

Moore's Law

Gordon Moore, founder of Intel Corporation, observed that the density of integrated circuits seemed to double every couple of years. This observation is referred to as "Moore's Law." Of course, it is not a natural law, like the law of gravity. Instead, it is an empirical observation of the progress of engineering and a challenge to engineers to continue their innovation. In 1965, Moore predicted that this exponential growth would continue for quite some time. That it has continued for more than 40 years is one of the great marvels of engineering. No other effort in history has sustained anything like this growth rate.

achieve the same effect as continued increases in raw processor speed. And the same technology improvements that make computers faster also make them cheaper.

The rapid increase in processing power means that inventions move out of labs and into consumer goods very quickly. Robot vacuum cleaners and self-parking vehicles were possible in theory a decade ago, but now they have become economically feasible. Tasks that today seem to require uniquely human skills are the subject of research projects in corporate or academic laboratories. Face recognition and voice recognition are poised to bring us new inventions, such as telephones that know who is calling and surveillance cameras that don't need humans to watch them. The power comes not just from the bits, but from being able to do things with the bits.

Koan 5: More of the Same Can Be a Whole New Thing

Explosive growth is exponential growth—doubling at a steady rate. Imagine earning 100% annual interest on your savings account—in 10 years, your money would have increased more than a thousandfold, and in 20 years, more than a millionfold. A more reasonable interest rate of 5% will hit the same growth points, just 14 times more slowly. Epidemics initially spread exponentially, as each infected individual infects several others.

When something grows exponentially, for a long time it may seem not to be changing at all. If we don't watch it steadily, it will seem as though something discontinuous and radical occurred while we weren't looking.

That is why epidemics at first go unnoticed, no matter how catastrophic they may be when full-blown. Imagine one sick person infecting two healthy people, and the next day each of those two infects two others, and the next day after that each of those four infects two others, and so on. The number of newly infected each day grows from two to four to eight. In a week, 128 people come down with the disease in a single day, and twice that number are now sick, but in a population of ten million, no one notices. Even after two weeks, barely three people in a thousand are sick. But after another week, 40% of the population is sick, and society collapses

Exponential growth is actually smooth and steady; it just takes very little time to pass from unnoticeable change to highly visible. Exponential growth of anything can suddenly make the world look utterly different than it had been. When that threshold is passed, changes that are "just" quantitative can look qualitative.

Another way of looking at the apparent abruptness of exponential growth—its explosive force—is to think about how little lead time we have to respond to it. Our hypothetical epidemic took three weeks to overwhelm the population. At what point was it only a half as devastating? The answer is *not* "a week and a half." The answer is *on the next to last day*. Suppose it took a week to develop and administer a vaccine. Then noticing the epidemic after a week and a half would have left ample time to prevent the disaster. But that would have required understanding that there *was* an epidemic when only 2,000 people out of ten million were infected.

The information story is full of examples of unperceived changes followed by dislocating explosions. Those with the foresight to notice the explosion just a little earlier than everyone else can reap huge benefits. Those who move a little too slowly may be overwhelmed by the time they try to respond. Take the case of digital photography.

In 1983, Christmas shoppers could buy digital cameras to hook up to their IBM PC and Apple II home computers. The potential was there for anyone to see; it was not hidden in secret corporate laboratories. But digital photography did not take off. Economically and practically, it couldn't. Cameras were too bulky to put in your pocket, and digital memories were too small to hold many images. Even 14 years later, film photography was still a robust industry. In early 1997, Kodak stock hit a record price, with a 22% increase in quarterly profit, "fueled by healthy film and paper sales...[and] its motion picture film business," according to a news report. The company raised its dividend for the first time in eight years. But by 2007, digital memories had become huge, digital processors had become fast and compact, and both were cheap. As a result, cameras had become little computers. The company that was once synonymous with photography was a shadow of its former self. Kodak announced that its employee force would be cut to 30,000, barely a fifth the size it was during the good times of the late 1980s. The move would cost the company more than $3 billion. Moore's Law moved faster than Kodak did.

In the rapidly changing world of bits, it pays to notice even small changes, and to do something about them.

Koan 6: Nothing Goes Away

2,000,000,000,000,000,000,000.

That is the number of bits that were created and stored away in 2007, according to one industry estimate. The capacity of disks has followed its own version of Moore's Law, doubling every two or three years. For the time being at least, that makes it possible to save everything though recent projections suggest that by 2011, we may be producing more bits than we can store.

In financial industries, federal laws now *require* massive data retention, to assist in audits and investigations of corruption. In many other businesses, economic competitiveness drives companies to save everything they collect and to seek out new data to retain. Wal-Mart stores have tens of millions of transactions every day, and every one of them is saved—date, time, item, store, price, who made the purchase, and how—credit, debit, cash, or gift card. Such data is so valuable to planning the supply chain that stores will pay money to get more of it from their customers. That is really what supermarket loyalty cards provide—shoppers are supposed to think that the store is granting them a discount in appreciation for their steady business, but actually the store is paying them for information about their buying patterns. We might better think of a privacy tax—we pay the regular price *unless* we want to keep information about our food, alcohol, and pharmaceutical purchases from the market; to keep our habits to ourselves, we pay extra.

The massive databases challenge our expectations about what will happen to the data about us. Take something as simple as a stay in a hotel. When you check in, you are given a keycard, not a mechanical key. Because the keycards can be deactivated instantly, there is no longer any great risk associated with losing your key, as long as you report it missing quickly. On the other hand, the hotel now has a record, accurate to the second, of every time you entered your room, used the gym or the business center, or went in the back door after-hours. The same database could identify every cocktail and steak you charged to the room, which other rooms you phoned and when, and the brands of tampons and laxatives you charged at the hotel's gift shop. This data might be merged with billions like it, analyzed, and transferred to the parent company, which owns restaurants and fitness centers as well as hotels. It might also be lost, or stolen, or subpoenaed in a court case.

The ease of storing information has meant asking for more of it. Birth certificates used to include just the information about the child's and parents' names, birthplaces, and birthdates, plus the parents' occupations. Now the electronic birth record includes how much the mother drank and smoked during her pregnancy, whether she had genital herpes or a variety of other medical conditions, and both parents' social security numbers. Opportunities for research are plentiful, and so are opportunities for mischief and catastrophic accidental data loss.

And the data will all be kept forever, unless there are policies to get rid of it. For the time being at least, the data sticks around. And because databases are intentionally duplicated—backed up for security,

> *The data will all be kept forever, unless there are policies to get rid of it.*

or shared while pursuing useful analyses—it is far from certain that data can ever be permanently expunged, even if we wish that to happen. The Internet consists of millions of interconnected computers; once data gets out, there is no getting it back. Victims of identity theft experience daily the distress of having to remove misinformation from the record. It seems never to go away.

Koan 7: Bits Move Faster Than Thought

The Internet existed before there were personal computers. It predates the fiber optic communication cables that now hold it together. When it started around 1970, the ARPANET, as it was called, was designed to connect a handful of university and military computers. No one imagined a network connecting tens of millions of computers and shipping information around the world in the blink of an eye. Along with processing power and storage capacity, networking has experienced its own exponential growth, in number of computers interconnected and the rate at which data can be shipped over long distances, from space to earth and from service providers into private homes.

The Internet has caused drastic shifts in business practice. Customer service calls are outsourced to India today not just because labor costs are low there. Labor costs have *always* been low in India, but international telephone calls used to be expensive. Calls about airline reservations and lingerie returns are answered in India today because it now takes almost no time and costs almost no money to send to India the bits representing your voice. The same principle holds for professional services. When you are X-rayed at your local hospital in Iowa, the radiologist who reads the X-ray may be half a world away. The digital X-ray moves back and forth across the world faster than a physical X-ray could be moved between floors of the hospital. When you place an order at a drive-through station at a fast food restaurant, the person taking the order may be in another state. She keys the order so it appears on a computer screen in the kitchen, a few feet from your car, and you are none the wiser. Such developments are causing massive changes to the global economy, as industries figure out how to keep their workers in one place and ship their business as bits.

In the bits world, in which messages flow instantaneously, it sometimes seems that distance doesn't matter at all. The consequences can be startling. One of us, while dean of an American college, witnessed the shock of a father receiving condolences on his daughter's death. The story was sad but familiar, except that this version had a startling twist. Father and daughter were both in Massachusetts, but the condolences arrived from half-way around the world before the father had learned that his daughter had died. News, even the most intimate news, travels fast in the bits world, once it gets out. In the fall of 2007, when the government of Myanmar suppressed protests by Buddhist monks, television stations around the world showed video clips taken by cell phone, probably changing the posture of the U.S. government. The Myanmar rebellion also shows the power of information control when information is just bits. The story dropped off the front page of the newspapers once the government took total control of the Internet and cell phone towers.

The instantaneous communication of massive amounts of information has created the misimpression that there is a place called "Cyberspace," a land without frontiers where all the world's people can be interconnected as though they were residents of the same small town. That concept has been decisively refuted by the actions of the world's courts. National and state borders still count, and count a lot. If a book is bought online in England, the publisher and author are subject to British libel laws rather than those of the homeland of the author or publisher. Under British law, defendants have to prove their innocence; in the U.S., plaintiffs have to prove the guilt of the defendants. An ugly downside to the explosion of digital information and its movement around the world is that information may become less available even where it would be legally protected (we return to this subject in Chapter 7, "You Can't Say That on the Internet"). Publishers fear "libel tourism"—lawsuits in countries with weak protection of free speech, designed to intimidate authors in more open societies. It may prove simpler to publish only a single version of a work for sale everywhere, an edition omitting information that might somewhere excite a lawsuit.

Good and Ill, Promise and Peril

The digital explosion has thrown a lot of things up for grabs and we all have a stake in who does the grabbing. The way the technology is offered to us, the way we use it, and the consequences of the vast dissemination of digital information are matters not in the hands of technology experts alone. Governments and corporations and universities and other social institutions have a say. And ordinary citizens, to whom these institutions are accountable, can influence their decisions. Important choices are made every year, in government offices and legislatures, in town meetings and police stations, in the corporate offices of banks and insurance companies, in the purchasing departments of chain stores and pharmacies. We all can help raise the level of discourse and understanding. We can all help ensure that technical decisions are taken in a context of ethical standards.

We offer two basic morals. The first is that information technology is inherently neither good nor bad—it can be used for good or ill, to free us or to shackle us. Second, new technology brings social change, and change comes with both risks and opportunities. All of us, and all of our public agencies and private institutions, have a say in whether technology will be used for good or ill and whether we will fall prey to its risks or prosper from the opportunities it creates.

Technology Is Neither Good nor Bad

Any technology can be used for good or ill. Nuclear reactions create electric power and weapons of mass destruction. The same encryption technology that makes it possible for you to email your friends with confidence that no eavesdropper will be able to decipher your message also makes it possible for terrorists to plan their attacks undiscovered. The same Internet technology that facilitates the widespread distribution of educational works to impoverished students in remote locations also enables massive copyright infringement. The photomanipulation tools that enhance your snapshots are used by child pornographers to escape prosecution.

The key to managing the ethical and moral consequences of technology while nourishing economic growth is to *regulate the use* of technology without *banning or restricting its creation.*

It is a marvel that anyone with a smart cell phone can use a search engine to get answers to obscure questions almost anywhere. Society is rapidly being freed from the old limitations of geography and status in accessing information.

The same technologies can be used to monitor individuals, to track their behaviors, and to control what information they receive. Search engines need not return unbiased results. Many users of web browsers do not realize that the sites they visit may archive their actions. Technologically, there could be a record of exactly what you have been accessing and when, as you browse a library or bookstore catalog, a site selling pharmaceuticals, or a service offering advice on contraception or drug overdose. There are vast opportunities to use this information for invasive but relatively benign purposes, such as marketing, and also for more questionable purposes, such as blacklisting and blackmail. Few regulations mandate disclosure that the information is being collected, or restrict the use to which the data can be put. Recent federal laws, such as the USA PATRIOT Act, give government agencies sweeping authority to sift through mostly innocent data looking for signs of "suspicious activity" by potential terrorists—and to notice lesser transgressions, such as Governor Spitzer's, in the process. Although the World Wide Web now reaches into millions of households, the rules and regulations governing it are not much better than those of a lawless frontier town of the old West.

BLACKLISTS AND WHITELISTS

In the bits world, providers of services can create blacklists or whitelists. No one on a blacklist can use the service, but everyone else can. For example, an auctioneer might put people on a blacklist if they did not pay for their purchases. But service providers who have access to other information about visitors to their web sites might use undisclosed and far more sweeping criteria for blacklisting. A whitelist is a list of parties to whom services are available, with everyone else excluded. For example, a newspaper may whitelist its home delivery subscribers for access to its online content, allowing others onto the whitelist only after they have paid.

New Technologies Bring Both Risks and Opportunities

The same large disk drives that enable anyone with a home computer to analyze millions of baseball statistics also allow anyone with access to confidential information to jeopardize its security. Access to aerial maps via the Internet makes it possible for criminals to plan burglaries of upscale houses, but technologically sophisticated police know that records of such queries can also be used to solve crimes.

Even the most un-electronic livelihoods are changing because of instant worldwide information flows. There are no more pool hustlers today—journeymen wizards of the cue, who could turn up in pool halls posing as out-of-town bumpkins just looking to bet on a friendly game, and walk away with big winnings. Now when any newcomer comes to town and cleans up, his name and face are on `AZBilliards.com` instantly for pool players everywhere to see.

Social networking sites such as facebook.com, myspace.com, and match.com have made their founders quite wealthy. They have also given birth to many thousands of new friendships, marriages, and other ventures. But those pretending to be your online friends may not be as they seem. Social networking has made it easier for predators to take advantage of the naïve, the lonely, the elderly, and the young.

In 2006, a 13-year-old girl, Megan Meier of Dardenne Prairie, Missouri, made friends online with a 16-year-old boy named "Josh." When "Josh" turned against her, writing "You are a bad person and everybody hates you.... The world would be a better place without you," Megan committed suicide. Yet Josh did not exist. Josh was a MySpace creation—but of whom? An early police report stated that the mother of another girl in the neighborhood acknowledged "instigating" and monitoring the account. That woman's lawyer later blamed someone who worked for his client. Whoever may have sent the final message to Megan, prosecutors are having a hard time identifying any law that might have been broken. "I can start MySpace on every single one of you and spread rumors about every single one of you," said Megan's mother, "and what's going to happen to me? Nothing."

Along with its dazzling riches and vast horizons, the Internet has created new manifestations of human evil—some of which, including the cyber-harassment Megan Meier suffered, may not be criminal under existing law. In a nation deeply committed to free expression as a legal right, which Internet evils should be crimes, and which are just wrong?

Vast data networks have made it possible to move work to where the people are, not people to the work. The results are enormous business opportunities for entrepreneurs who take advantage of these technologies and new enterprises around the globe, and also the other side of the coin: jobs lost to outsourcing.

The difference every one of us can make, to our workplace or to another institution, can be to ask a question at the right time about the risks of some new technological innovation—or to point out the possibility of doing something in the near future that a few years ago would have been utterly impossible.

We begin our tour of the digital landscape with a look at our privacy, a social structure the explosion has left in shambles. While we enjoy the benefits of ubiquitous information, we also sense the loss of the shelter that privacy once gave us. And we don't know what we want to build in its place. The good and ill of technology, and its promise and peril, are all thrown together when information about us is spread everywhere. In the post-privacy world, we stand exposed to the glare of noonday sunlight—and sometimes it feels strangely pleasant.

Introduction to Abstraction

This activity introduces the concept that abstractions built upon binary sequences can be used to represent all digital data. It also introduces the idea the computing has global impacts. It focuses, in part, on the following learning objectives:

- The student can develop an abstraction.

Introduction

In this course, there's both a practical component, such as using SNAP/BYOB in the lab, and a "big ideas" component, in lecture and discussion. One of the things we want you to take away from this course is to know that there's more to computer science than just writing computer programs, although of course we use those other things to help in writing programs. So we'll talk about some aspects of the social context of computing, some of the theoretical explorations of the limits of computation, and some important moments in the history of computer science – for example, you'll learn in a few weeks how one of the first computer scientists more or less single-handedly won World War II for the good guys.

Abstraction

Abstraction is arguably the central idea of all of computer science. Computer programming is easy, as long as the programs are small. What's hard isn't the programming, but the keeping track of details in a huge program. The solution is **chunking**, or **layering** – two metaphors for abstraction.

Practical Abstraction Example

The classic example is thinking about a car. Cars are made of nuts, bolts, metal rods, big metal blocks, rubber or paper gaskets, plastic containers for fluids, rivets, wires, and so on. (Each piece of metal is further made of atoms, which are made of electrons, protons, and neutrons, which are made of quarks, and so on down.) But if you're trying to repair a car, you don't think in those terms; if you did, you'd never find where the problem is. Instead you think about the engine, the alternator, the fuel injectors, the brakes, the transmission, and so on. *That's* abstraction.

The march of technological progress is, at least in part, a march toward greater and greater abstraction. *Each step reduces the extent to which people have to think about details*. Sticking with cars as the example, in the early days, every driver had to be at least something of a mechanic, knowing how to deal with the rather frequent failures of the machinery. Before automatic transmissions, only people with some understanding of gear ratios could drive. (In the really early days, they couldn't downshift without mastering the skill of double-clutching.)

The automatic transmission made possible an enormous abstraction. All the complexity of the machinery that makes a car work was hidden under the surface of a very simple model: You push the pedal on the right and the car speeds up; you push the pedal on the left and it slows down. Suddenly just about anyone could drive a car.

Of course, the widespread use of cars has turned out to be a mixed blessing. Cars are one of the main causes of pollution and global warming. Computers, too, have their downsides, which we'll be discussing later. Many historians of science stay away from the word "progress," which I used two paragraphs back, because of its implicit suggestion that the development of new technology is always good. But before we can criticize technology we should understand something about how it works, and abstraction is a very powerful organizing idea to describe the mechanism.

Those two pedals, the gas pedal and the brake, are an **interface**, also known as an **abstraction barrier**. On the driver's side of the abstraction, what matters is the *behavior* provided by this interface. Push this one to speed up, that one to slow down. Once that interface became standardized, further technical development has dramatically changed what happens up in the engine compartment. Originally, the gas pedal mechanically pushed a lever controlling a valve that determined the rate at which gasoline could flow into the engine. More gasoline, bigger explosions inside the engine, more power, so more speed. Today, the gas pedal doesn't really do anything mechanically, except provide an input to a computer inside the car, whose job is to control the fuel injection system. Your input is combined with other information about the car's environment to operate smaller valves, one per cylinder, that control the gas/air mixture more precisely.

The brake pedal has had a similar history. Originally, your foot directly provided the power to push the brake pads against the wheels. Then a new mechanism was developed, preserving the interface – push here to slow down – but now using the pressure from your foot to operate a hydraulic system that does the hard work of pressing the pads against the wheels. But there was one important difference. The first "power brakes," like the modern gas pedal, completely eliminated the mechanical linkage between the pedal and the actual brakes. But after a few accidents in which people couldn't stop their cars because the engine died, this design was modified. Today you have "power assisted brakes," which means that your foot both operates a hydraulic cylinder and also directly puts pressure on the brakes. If the engine fails, you have to push a lot harder to stop the car, but at least it's possible. The latest development, anti-lock brakes, actually lets a computer in the car override your pressure on the brake pedal if you are in danger of putting the car into a skid by trying to stop too abruptly.

But the point is that drivers who aren't particularly interested in cars don't have to know any of this. All they have to know is that you push the pedal on the right to speed up, and the one on the left to slow down! This *interface* has survived through several generations of underlying technology, because it's a good interface – simple but expressive. Car engineers could have made each generation of new technology more visible to drivers, with lots of knobs and switches and readouts, but they wisely refrained, and stayed with the abstraction – the interface – developed a century ago.

Abstraction in Computer Programming

How does abstraction work in computer programming? We'll be revisiting this question all semester. But, for a starting point, think about the blocks in the BYOB menu. Each block names a simple intention, comparable to "speed up" or "slow down" in a car. But the mechanisms that make those blocks do their job are actually doing very complicated detail work. For example, "move 10 steps," above the abstraction barrier, just tells Scratch to move over a little. But BYOB isn't a real image; it's a collection of colored dots drawn on a computer screen. "Move 10 steps" really means to erase each of those dots, then redraw them all in a different position. And the "glide" block is even more complicated; it has to erase and redraw Scratch many times, moving just a tiny bit each time.

But if the abstraction is working well, you're not thinking about tiny dots of light at all, while working with Scratch/BYOB. You're moving a cat!

We will begin with BYOB, a visual programming language. You can take a complicated sequence of actions, wrap them up, and present them to another user – or to yourself thinking at a higher level of abstraction – as a new interface, allowing you to command a new behavior without having to know anything about the detailed implementation.

Material from Dr. Brian Harvey, UC Berkeley

More on Abstraction

This introduces the concept that *abstractions built upon binary sequences can be used to represent all digital data*. It also introduces the idea the *computing has global impacts*. It focuses, in part, on the following learning objectives:

- 5: The student can describe the combination of abstractions used to represent data.
- 6: The student can explain how binary sequences are used to represent digital data.
- 28: The student can analyze how computing affects communication, interaction, and cognition.
- 29: The student can connect computing with innovations in other fields.
- 30: The student can analyze the beneficial and harmful effects of computing.
- 31: The student can connect computing within economic, social, and cultural contexts.

Bits as the Natural Unit of Information

This was the seminal insight of Claude Shannon (http://en.wikipedia.org/wiki/Claude_Shannon), founder of information and communication theory. As this chapter illustrates all information stored on computers (documents, photos, songs, etc.) and communicated through the Internet (email, blogs, twitter posts, etc.) are represented as bits, zeros and ones.

Digital data are represented in discrete *bits* (**B**inary dig**IT**s). Analog data are represented as a continuous quantity. Think of the difference between a digital watch that displays hours, minutes, seconds, and tenths of a second, and an *analog clock*, with hour hand, minute hand, and a sweep second hand.

Think of the difference in *fidelity* between digital vs. analog recordings (http://www.howstuffworks.com/analog-digital3.htm).

Moore's Law

Moore's law (http://en.wikipedia.org/wiki/Moore%27s_law) is the observation that the number of transistors that could be packed onto a integrated circuit seemed to double ever two years or so. This remarkable trend about exponential growth (http://en.wikipedia.org/wiki/File:Exponential.svg) in chip density has proved to be true for 40 years now.

Someone offers you a summer job and offers you two payment schemes: (1) $10 per hour for 40 hours per week for 30 days or (2) One cent on day 1, two cents and day two, four cents on day three and on (doubling each day) for 30 days. Which pay would you choose? Click here to see how exponential growth affects this question (http://www.cs.trincoll.edu/~ram/aitalk/twos.txt).

It's All Just Bits

Binary digits (bits) correspond naturally to electronic circuits where 1 represents 'on' and 0 represents 'off'. In computers, binary sequences are used to represent all kinds of data: text, numbers, images, sounds, …, everything.

Depending on the context, the same string of bits can represent different types of information. These are all examples of **abstraction** at work:

- The sequence `0100 0001` can represent the binary numeral (http://en.wikipedia.org/wiki/Binary_numeral_system) for the decimal value 65 when it occurs calculator.
- The sequence `0100 0001` can represent the ASCII letter (http://www.ascii.cl/) 'A' when it occurs in an email message.

- The sequence `0100 0001` can represent the amount of Red in the RGB color model (http://en.wikipedia.org/wiki/RGB_color_model) when it occurs in an image.
- The sequence `0100 0001` can represent the MOV B,C machine operation on the Intel 8080 8-bit processor (http://nemesis.lonestar.org/computers/tandy/software/apps/m4/qd/opcodes.html)

A Little Bit About Bits

A **byte** is an 8-bit sequence. Historically an 8-bit byte was used to represent a single character in computer memory.

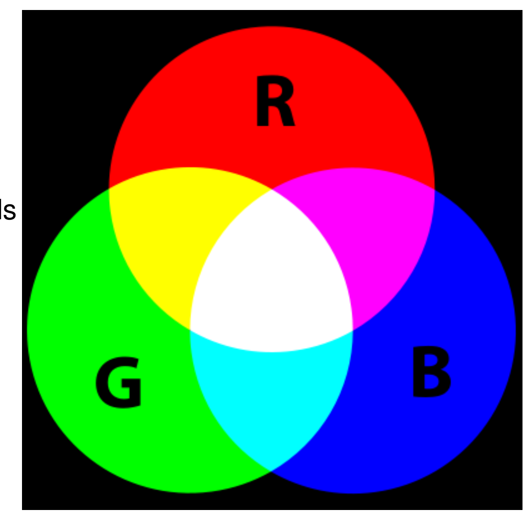
The length of a binary sequence – a sequence of 0s and 1s – determines the number of different sequences that can be generated and therefore the number of different things that can be represented by such a sequence.

For example, with 1 bit, you can have two different sequences, 0 or 1, which can stand for two different colors, say, 0 stands for white and 1 for black. With 2 bits, you can have four different sequences, 00, 01, 10, or 11, which can represent four different colors, white, black, red, green. And so on.

In general, an n-bit sequence can represent 2^n different things. Here’s how:

Number of Bits	Number of Things	Representations
1	2 (ie 2^1)	0,1
2	4 (ie 2^2)	00,01,10,11
3	8 (ie 2^3)	000,001,010,011,100,101,110,111
4	16 (ie 2^4)	
5	32 (ie 2^5)	
6	64 (ie 2^6)	
7	128 (ie 2^7)	
8	256 (ie 2^8)	

Data Abstraction: How Colors Are Represented in Bits



The RGB model (http://en.wikipedia.org/wiki/RGB_color_model) adds together 3 primary colors, Red, Green, and Blue, where each color component (R,G,B) is represented as an 8-bit **byte**.

If there are 256 (2^8) possible values for each of R,G,B, the triplet can represent 256 × 256 × 256 = 16,777,216 different colors, which corresponds well to the number of colors the human eye can distinguish.

- (R, G, B)
- (65,65,65) = Grey
- (255,0,0) = Red
- (0,255,0) = Green
- (0,0,255) = Blue
- (65,0,0) = Brown

Try to mix your own colors: Colorschemer.com (http://www.colorschemer.com/online.html)

Perfection is Normal

Because they are based on strings of discrete bits, *digital* (as opposed to *analog*) copies are perfect copies.

Computer scientists and engineers have developed effective error detection (http://en.wikipedia.org/wiki/Parity_bit) and error correction schemes to insure accurate data representation and communication.

Example: Parity Bit Error Detection

Suppose you are sending a stream of data to a server. By adding a parity bit, you enable to the server to detect some basic transmission errors. For example, if the server expects that every byte will contain an **even number of 1s** and it detects a byte such as `0001 0101` with an odd number of 1s, it can tell that an error occured. Perhaps the user meant to send `0000 0101` but one of the bits was flipped from 0 to 1 during transmission.

A **parity bit** is a bit that is added as the leftmost bit of a bit string to ensure that the number of bits that are 1 in the bit string are even or odd.

To see how this works, suppose our data are stored in strings containing 7 bits.

In an **even parity scheme** the eighth bit, the parity bit, is set to 1 if the number of 1s in the 7 data bits is odd, thereby making the number of 1s in the 8-bit byte an even number. It is set to 0 if the number of 1s in the data is even.

In an **odd parity scheme** the eighth bit, the parity bit, is set to 1 if the number of 1s in the 7 data bits is even, thereby making the number of 1s in the 8-bit byte an odd number. It is set to 0 if the number of 1s in the data is odd.

The following table summarize this approach.

Data Bits (7)	Even Parity (even number 1s)	Odd Parity (odd number 1s)
`000 0000` (0 1s)	`0000 0000`	`1000 0000`
`011 0010` (3 1s)	`1011 0010`	`0011 0010`
`011 0011` (4 1s)	`0011 0011`	`1011 0011`
`011 0111` (5 1s)	`1011 0111`	`0011 0111`

Question: What would happen in this scheme if 2 bits were switched from 1 to 0 or 0 to 1?

Quiz Yourself

To see if you understand these concepts, try the following quizzes. If you can get ten-in-a-row correct, that's a pretty good indication that you get it.

- Parity Error Detection I (http://www.cs.trincoll.edu/%7Eram/q/110/parity-error-detection.html)
- Parity Error Detection II (http://www.cs.trincoll.edu/%7Eram/q/110/parity-error-detection-2.html)

Variables and Abstraction

When you program you will create variables to make the script much more functional – i.e., the user could change the value of the variable.

This is an important example of **abstraction** at work – in this case, we are letting an *abstract symbol* (the variable) represent or stand for something else (its value, 2 or 8).

What is a Variable?

A *variable* in a computer program is a *symbol* that represents a *memory location* where a piece of data can be stored. You can think of a computer memory as a large array of numbered mail boxes, where the numbers represent the address of the memory location. In this example, there are 8 memory locations numbered 17-24 (in decimal) and 10001 - 11000 (in binary). None of the locations have any value stored in them yet.

17	18	19	20	21	22	23	24
10001	10010	10011	10100	10101	10110	10111	11000

When you define a *variable*, you are giving name to a memory location.

name	score	wins	20	21	22	23	24
10001	10010	10011	10100	10101	10110	10111	11000
Joe	8	2					

What is a Value?

In a computer program a *value* is a symbol of a piece of *data*. For example, the numeral '8' represents the number 8. Values (data) are stored in the computer's memory locations.

Symbols like the numeral '8' are also examples of *abstractions*. The numeral '8' and the word 'eight' and the binary string '1000' are all symbols that represent the number 8, which is itself an abstract concept of 8 things. For example, they can all be used to refer to the number of little circles here: o o o o o o o o.

So, in a computer program we use abstract symbols to represent both *variables* and *values*.

Values vs. Variables

It's important to distinguish between the variable's *name* (e.g., score or name) from the *value* that it represents – i.e, from the value that it is storing in its memory location (e.g., **8** or **Joe**).

Material from Dr. Ralph Morelli, Trinity College

Binary, Hex, and Decimal Numbers

This activity addresses the concept of abstraction. It focuses, in part, on the following learning objectives:

- 5b. Explanation of how number bases, including binary and decimal, are used for reasoning about digital data.

Introduction

As we've discussed, binary sequences are an important way to represent digital data. Our number system, which we call the *decimal* system, is a *base-10* number system. It uses 10 digits – 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 – and the places in a decimal number are based on powers of 10 – i.e., the ones place (10^0), tens place (10^1), hundreds place (10^2), and so on.

The *binary* system is a *base-2* system. It uses just 2 digits – 0 and 1 – and the places in the number are based on the powers of two. So we have the ones place (2^0), the twos place (2^1), the fours place (2^2), and so on.

Because binary numbers can get very long, computer scientists use other number systems based on powers of 2 that make it easier to represent digital data. One of these is the *hexadecimal* system, which is a *base-16* system. It has 16 digits – 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. And the places in a hexadecimal number are based on powers of 16. So we have the ones place (16^0), the sixteens place, (16^1), the 256-place (16^2) and so on.

In this homework you will watch a short (10 minute) video about binary numbers and learn how to convert.

Khan Academy video on binary

- Binary numbers (http://www.khanacademy.org/video/binary-numbers?playlist=Pre-algebra)

Number Conversions

- Read about Number Conversions (http://www.cstutoringcenter.com/tutorials/general/convert.php) from CS Tutoring Center

Quiz Yourself

To reinforce your understanding of these concepts, try the following quizzes. If you can get ten-in-a-row correct, that's a pretty good indication that you get it.

- Binary to Decimal (http://www.cs.trincoll.edu/~ram/q/110/binary-to-decimal.html)
- Decimal to Binary (http://www.cs.trincoll.edu/~ram/q/110/decimal-to-binary.html)
- Hexadecimal to Decimal (http://www.cs.trincoll.edu/~ram/q/110/hex-to-decimal.html)
- Decimal to Hexadecimal (http://www.cs.trincoll.edu/~ram/q/110/decimal-to-hex.html)
- Binary to Hexadecimal (http://www.cs.trincoll.edu/~ram/q/110/binary-to-hex.html)
- Hexadecimal to Binary (http://www.cs.trincoll.edu/~ram/q/110/hex-to-binary.html)

Material from Dr. Ralph Morelli, Trinity College

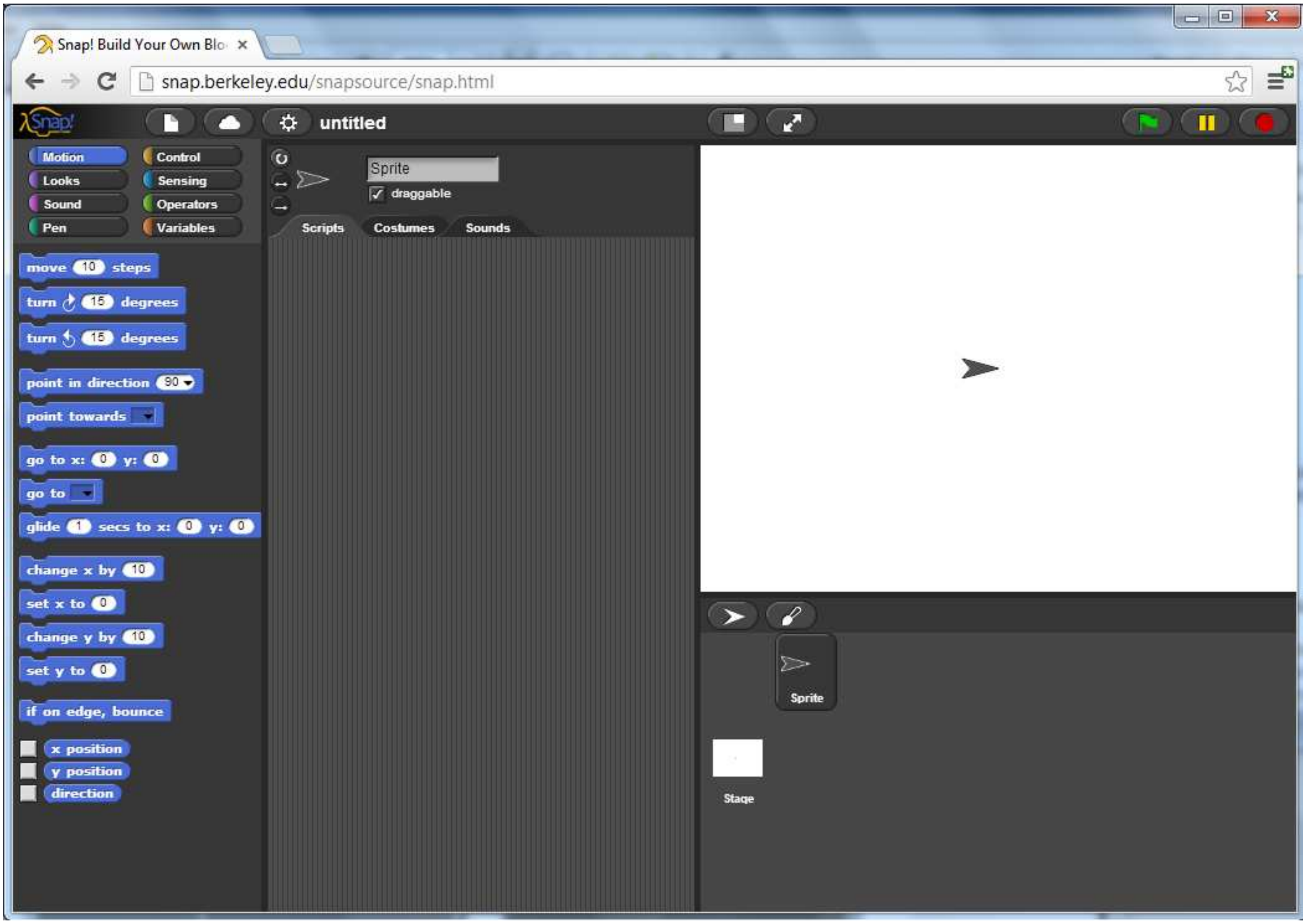
Conversion Exercise

1. Convert the following numbers to decimal notation.
    1. The binary number 111.
    2. The binary number 1011.
    3. The binary number 1011 1011.
    4. The hex number 61.
    5. The hex number DA.
    6. The hex number FEE.
2. Convert the following decimal numbers as indicated.
    1. Convert decimal 12 to binary.
    2. Convert decimal 44 to binary.
    3. Convert decimal 254 to hex.
    4. Convert decimal 16 to hex.
3. Challenge: Convert decimal 125 to octal (base 8) notation.

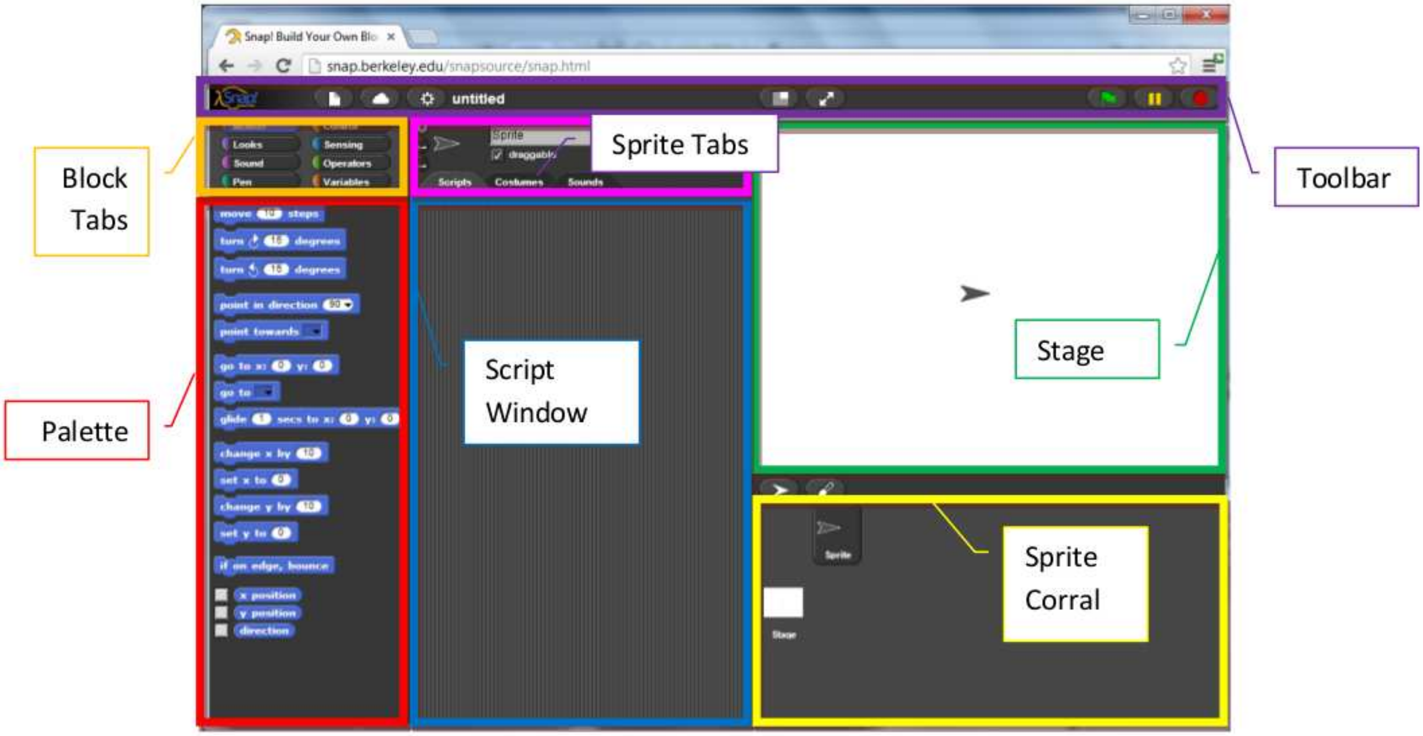
Lab: Welcome to Visual Programming

Let's open up SNAP at http://snap.berkeley.edu/run (http://snap.berkeley.edu/run)

You will see a screen like the one shown below. Explore the aspects of the user interface. Play around for a while and see if you can figure out the major components of the interface. In the next step, you will make your first project and explore further.



Let's first look at the IDE (Integrated Development Environment).



Make a sprite sing

For your first project, make a quick song! You will find the following **blocks** in the Sound tab useful; feel free to change the default numbers as you see fit.

While you are working on it, try to figure out how to connect and disconnect blocks, and how to remove a piece from inside a long script. Also, what do you think is the difference between these two blocks?

Hint: Try to use many copies of one of the blocks in a row, and hear the result. Do this for each block.

Meowing: One at a Time or in Unison?

With this brief introduction to the scratch interface, we begin to examine how sprites and blocks interact and affect one another. For example, the "play sound" blocks from earlier allow us to control when and how many sounds we here. Consider the difference between these two blocks?

If you set up a small script like this, how many meows do you you hear?

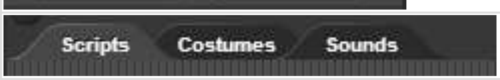
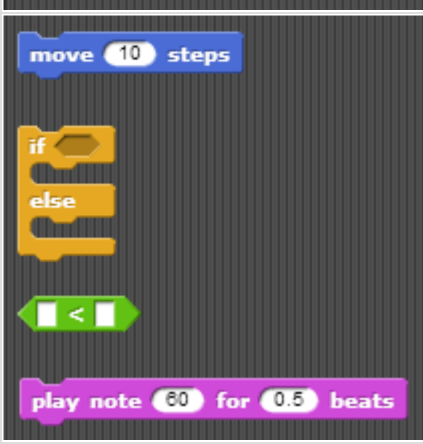
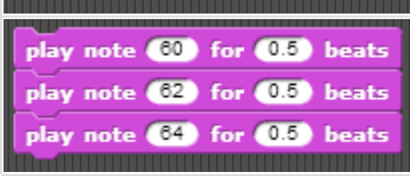
How about one like this: How many meows do you hear now?

Experiment with these blocks: 1) How about two “play sound (meow)” and then one “play sound (meow) until done”?

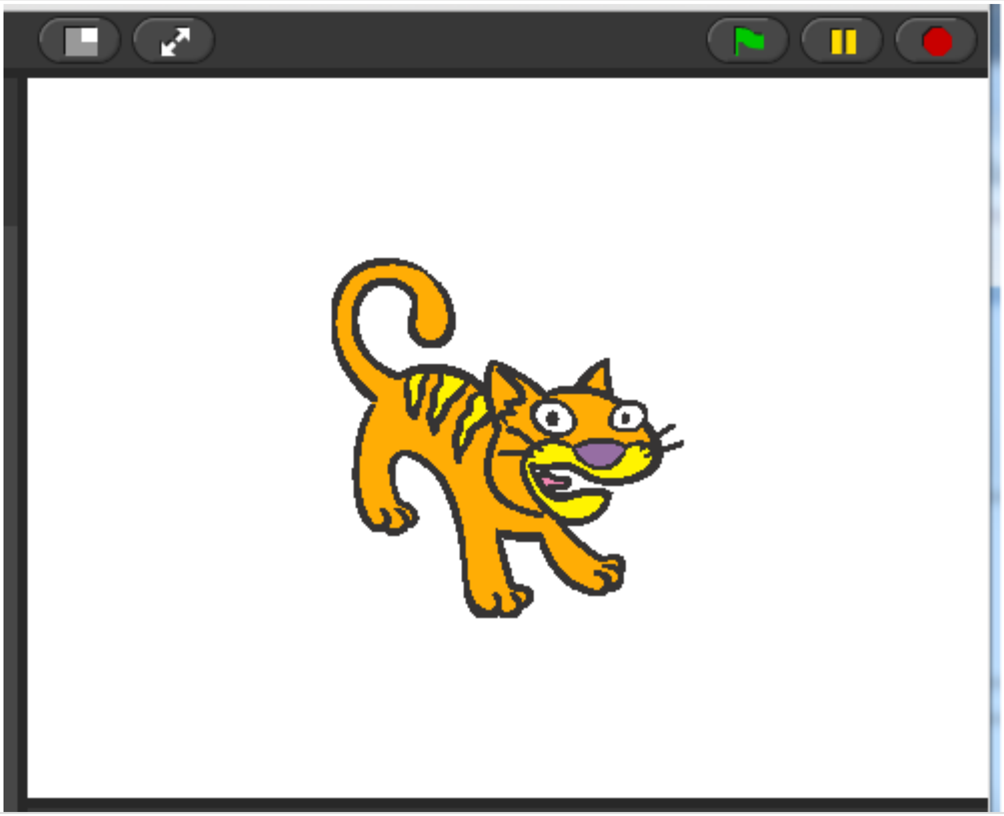
2) How about two “play sound (meow) until done” and then one “play sound (meow)”

Explain the difference between 1) and 2). Why did you hear a different amount of meows?

Some Starting Lingo

Term	Example/Description
Tabs (for blocks)	
Tabs (for sprite)	
Blocks	
Script	

Term	Example/Description
Sprite	
Costumes (Each sprite can have multiple costumes)	

Term	Example/Description
Stage	
Bug	A defect (or a problem) in the code or routine of a program.
User Interface	The place where interaction between humans and machines occurs. For exam, Windows, Scratch, the iPod touch screen, and even your keyboard are examples of user interaces.

Experiment with a Short Play

Try to make these scripts in SNAP! You will find that the Cat and the Duck have completely separate script areas. Click on each character to see their script area. Once you are done, press the green flag to start the short play.

Cat's Script Window

Duck's Script Window

A note about style

You will notice that we chose to name the messages that were broadcast so that it would help us keep track of what we were doing and what messages we were sending. We recommend that you do this in your projects!

Hints

To choose a new sprite, drag an image file into the Costume area. Click on the paint brush to create a new image of your own.

Try to figure out what the commands (whose images are on the left) and buttons (whose images are on the right) do. These will be helpful to get the characters to face each other.

Try It! Play

Once you have this working, change the sprites, and then change the script of the "play" so that each character says at least two additional lines.

Exporting Sprites

By this point, you have probably figured out how to save your projects, but you can also save individual Sprites separately. To save (or export) a Sprite, right-click on the sprite and select export this sprite. To load (or import) a Sprite, click on the icon with a folder next to New Sprite (circled in yellow in the image below) and select the Sprite that you want to add to your project.

Lights, Camera, Action

In this activity, you will create a movie or a play.

In general, your movie should have at least:

- One block that we did not use during lab. (Explore your blocks)
- Two characters not previously used.
- Ten (10) total broadcasts of messages. This can be five each, or six/four, etc.
- Add sounds (Sound Tab)

The activity is open to your creativity – tell a story, create characters, experiment with moving sprites around stage