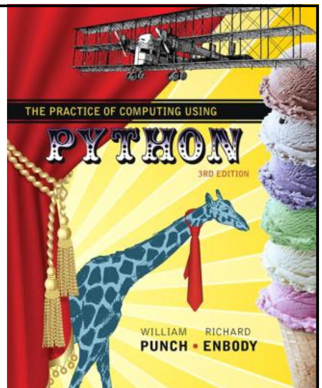
Chapter 0

The Study of Computer Science



PEARSON ALWAYS LEARNING

What is Computer Science?

 2

Definition

- computer science is a discipline that involves the understanding and design of computers and computational processes.

 3

Motivation

- a well-educated computer scientist should be able to
  - apply the fundamental concepts and techniques of
    - computation
    - algorithms
    - computer design
  - to a specific problem

 4

Motivation

- …including
  - detailing of specifications
  - analysis of the problem
  - provide a design that
    - functions as desired
    - has satisfactory performance
    - is reliable and maintainable
    - and meets desired cost criteria

 5

Aspects of Computer Science

- Theory of computation
- Computational Efficiency
- Algorithms and Data Structures
- Parallel Processing
- Software Engineering
- others

 6

Goals

- our goals are not to just write copious amounts of code, our goals are to
  - increase our problem solving skills
  - design good solutions to problems
  - test somehow how well they are indeed solutions to the problem
  - provide the solution as a readable document

Why is this Hard?

- computer science often difficult for students
- typical quote is "never have I worked so hard and gotten so low a grade"

An Analogy

- say that you have signed up to study French poetry (e.g., Marot) in the original French
- you have two problems
  - you don't speak French
  - you don't know much about poetry

Clement Marot (1496-1544)

Ma mignonne
Je vous donne
Le bon jour;
Le séjour
C'est prison.
Guérison
Recouvrez,
Puis ouvrez
Votre porte
Et qu'on sorte
Vitement,
Car Clément
Le vous mande.
Va, friane
De ta bouche,
Qui se couche
En danger
Pour mange
Confitures;
Si tu dures
Trop malade,
Couleur fade
Tu Prendras,
Et perdras
L'embonpoint.
Dieu te doint
Santé bonne,
Ma mignonne.

Literal Translation

My sweet/cute [one] (feminine)
I [to] you (respectful) give/bid/convey
The good day (i.e., a hello, i.e., greetings).
The stay/sojourn/visit (i.e., quarantine)
[It] is prison.
Cure/recovery/healing (i.e., [good] health)
Recover (respectful imperative),
[And] then open (respectful imperative)
Your (respectful) door,
And [that one (i.e., you (respectful)) should} go out
Fast[ly]/quick[ly]/rapid[ly],
For/because Clement
It (i.e., thusly) [to] you(respectful) commands/orders.
Go (familiar imperative), fond-one/enjoyer/partaker
Of your (familiar) mouth,
Who/which herself/himself/itself beds (i.e., lies down)
In danger;
For/in-order-to eat
Jams/jellies/confectionery.
If you (familiar) last (i.e., stay/remain)
Too sick/ill,
[A] color pale/faded/dull
You)familiar) will take [on],
And [you (familiar)] will waste/lose
The plumpness/stoutness/portliness (i.e., well-fed look).
[may] God [to] you (familiar) give/grant
Health good,
My sweet/cute [one] (feminine).

Decent Translation
by S. Jamar

My sweet dish,
You I wish
A good day.
Where you stay,
Is a jail.
Though so pale,
Leave your bed,
Regain red.
Ope' your door
Stay not, poor
Child; gain strength
And at length,
Steve does urge,
Please emerge.
Then go eat
Jam so sweet.
Lying ill
Means you will
become too thin -
Merely skin
Cov'ring bone;
Regretted tone.
Eat again,
Avoid the fen.
God grant thee
Be healthy.
This I wish,
My sweet dish.

How Does This Apply?

- you have two related problems
  - the syntax of French is something you have to learn
  - the semantics of poetry is something you have to learn
- you have two problems you have to solve at the same time

 13

Learning to Program

- you have to learn the syntax of a particular programming language
  - many details about the language, how to debug and use it
- you have to learn about problem solving and how to put it down on the computer
- there probably is no better way
- it's hard!

 14

Computers and Problem Solving

- both the problem and difficulty of computers
  - the promise (perhaps the hope) of computers is that, somehow, we can embed our own thoughts in them (to some extent we can)
  - the problem is the difficulty of doing so, and the stringent requirements, the real rigor, required to put simple thoughts into a working program

 15

Focus of Computer Science

- two foci for computer science
  - learning the difficult task of truly "laying out" a problem-solving task
  - providing tools to make this process as easy (though it will never be "easy") as possible
- your focus should be on problem-solving, and adding rigor/focus to your ability to do problem-solving

 16

Good Program (1)

- What makes a good program?
  - a program is a reflection of the writer and their thoughts
  - first, you must have some thoughts!
  - the difficulty for most people is to figure out what has to be done, the problem solving, before writing a program

 17

Rule 1

Think before you program!

 18

Good Program (2)

- it will be said repeatedly that the goal of a program is **not** to run, but to be read
  - actually, it's both
- a program communicates with other people as well
- it stands as a document to be read, repaired and, yes, run

Rule 2

A program is a human-readable essay on problem solving that also executes on a computer.

Why Python?

- this class utilizes the programming language known as Python
- why?

Why Python (1): Simpler

- Python is a simpler language
- simpler means
  - fewer alternatives (one way to do it)
  - better alternatives (easier to accomplish common tasks)
- this allows us to focus less on the language and more on problem solving

Why Python (2): Best Practices

- many of the best parts of other languages are included in Python
  - data structures (lists, dictionaries)
  - control (iteration, exceptions)
  - many packages for common tasks
- Python is often described as "batteries included"

Why Python(3): User Base

- while we want to (and will) teach the fundamentals of computer science, we want what you learn to be useful
- Python is open source
  - freely available
  - large user base constantly contributing
  - new packages available to meet changing needs

Why Python (4): Useful

- as a result, Python is more generally useful for getting work done
- one course in Python makes you a capable (though not expert) programmer
- can use available packages and your new skill to solve problems

 25

Computational Thinking

- having finished this course, we want you to have the following thought in your subsequent college career:
  - "Hey, I'll just write a program for that"
  - computational thinking
- Python allows this to happen more readily

 26

Is Python the Best Language?

- no, because there is no best language
- computer languages, like tools, are suited for different tasks
  - what's the best shovel? depends on what you are doing
- for introductory students, Python is a very good language

 27

What is a Computer?

 28

Computer as a Toaster

- 50 years ago when computers came into common existence, they were expensive, rare items used for research
- today they are as common as a toaster
  - likely more common, do you own a toaster?
- good to know a little bit about your toaster

 29

Computers Perform Computation

- kind of obvious, but a computer is something that does computation
- what's interesting is what counts as computation

 30

Human Brain

- in the 1960's, when computers were becoming more prevalent, they were commonly called "electronic brains"
- however, the brain as a computer has very little in common with the modern computer
  - except that they both do computation

Neuron

- the human brain consists of on the order of 100 billion (10^{11}) small cells called neurons
- neurons have a simple task: to act as a kind of switch

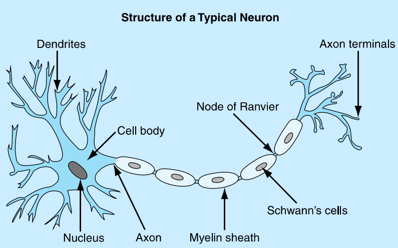


FIGURE 0.3 An annotated neuron. [SEER Training Modules, U.S. National Institutes of Health, National Cancer Institute]

Transmission of Signal

- a neuron "fires" and sends a signal when enough chemical transmitters accumulate at its dendrites
- when it fires, the signal, as a chemical/ electrical impulse, moves down the axon
- when the signal reaches the terminals, more chemical transmitters are released to other neurons

Interesting Facts

- signal transmission is very slow
  - millions times slower than a computer
- neurons require recovery time before they can fire again
- what causes a firing depends on the number of signals it receives at its dendrites and how fast they arrive
  - that is, it depends on the number of connections

People!

- computers used to refer to *people*
- in WWII, computers were people who did difficult calculations *by hand*, for things like ballistic tables

The Modern Computer

 37

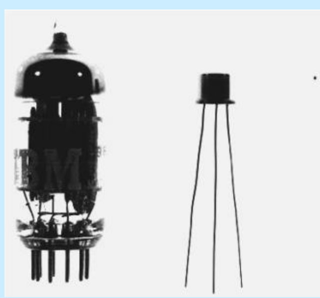
The Switch

- the basic component of most digital circuitry is nothing more complicated than a simple switch
- a switch's function is pretty obvious, said in a number of different ways
  – on or off
  – True or False
  – 1 or 0

 38

Electronic switch

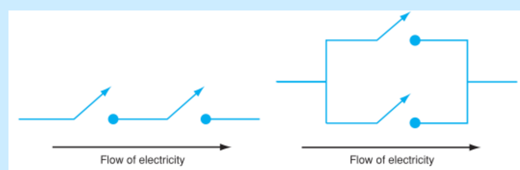
- early computers used vacuum tubes as switches
- later, transistors were used as substitutes



 39

Switches for Boolean Circuits

- switches can be used to construct more complicated functions, such as Boolean circuits (**`and`** on left, **`or`** on right)



 40

The Transistor is the Key

- a transistor is an electronic switch
  – if gate has current, then signal flows from source to sink
  – if gate has no current, no flow

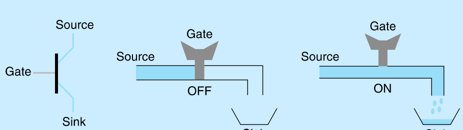


FIGURE 0.8 A diagram of a transistor and its equivalent "faucet" view.

 41

But a Very Interesting Switch

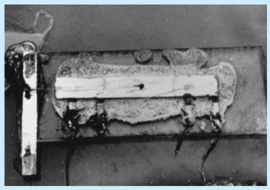
- transistors have three interesting features that have made them the fundamental element of the computer revolution
  – size
  – quantity
  – speed

 42

Size

- originally very large

Shockley transistor

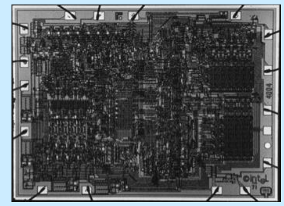


Kilby integrated circuit

 43

Intel's First CPU

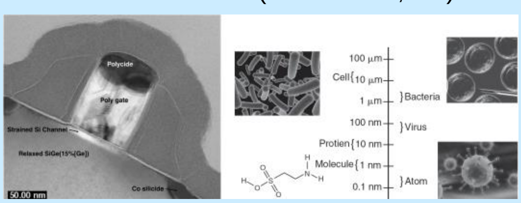
- by 1971, Intel had created a "computer on a chip"
  - 4004 microprocessor
  - size of a fingernail
  - 2300 transistors



 44

Today, *Really* Small!

- chips now have gates measured in billionths of a meter (nanometers, nm)



 45

Smaller Means More…

Year	Transistor Count	Model
1971	2,300	4004
1978	29,000	8086
1982	134,000	80286
1986	275,000	80386
1989	1,200,000	80486
1993	3,100,000	Pentium
1999	9,500,000	Pentium III
2001	42,000,000	Pentium 4
2007	582,000,000	Core 2 Quad
2011	2,600,000,000	10-core Westmere

TABLE 0.1 Transistor counts in Intel microprocessors, by year.

 46

GPUs, Too!

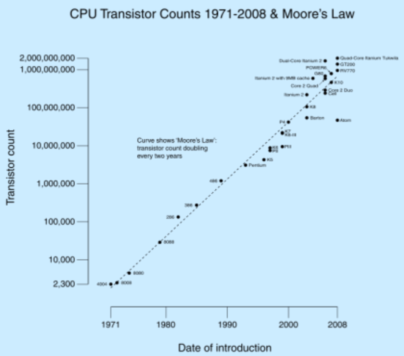
Processor	count	Date	Co.	Process	Area
G80	681,000,000	2006	NVIDIA	90 nm	480 mm²
RV770	956,000,000	2008	AMD	55nm	
GT200	1,400,000,000	2008	NVIDIA	55 nm	576 mm²
HD5800	2,154,000,000	2009	AMD	40 nm	334 mm²
GF100	2,900,000,000	(future)	NVIDIA	40 nm	

 47

Moore's Law

- Gordon Moore is one of the founders of the chip maker Intel
- in 1965, he observed (over that last 15 years or so) the growth rate of the number of transistors in a circuit
- made a famous prediction…

 48



Moore's Law

"The complexity for minimum component costs has increased at a rate of roughly a factor of two per year ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer"

Electronics Magazine 19 April 1965

What it Means

- roughly, since 1965, the number of transistors on a chip doubles every 18 months for approximately the same cost
- often quoted as the speed of a CPU doubling every 18 months for the same cost
- speed and density were related

How Fast?

- typical processor this day runs at GHz speeds (1 billion operations a second)

Year	CPU	Instructions/second	Clock Speed
1971	Intel 4004	1 MIPS	740 kHz
1972	IBM System/370	1 MIPS	?
1977	Motorola 68000	1 MIPS	8 MHz
1982	Intel 286	3 MIPS	12 MHz
1984	Motorola 68020	4 MIPS	20 MHz
1992	Intel 486DX	54 MIPS	66 MHz
1994	PowerPC 600s (G2)	35 MIPS	33 MHz
1996	Intel Pentium Pro	541 MIPS	200 MHz
1997	PowerPC G3	525 MIPS	233 MHz
2002	AMD Athlon XP 2400+	5,935 MIPS	2.0 GHz
2003	Pentium 4	9,726 MIPS	3.2 GHz
2005	Xbox360	19,200 MIPS	3.2 GHz
2006	PS3 Cell BE	10,240 MIPS	3.2 GHz
2006	AMD Athlon FX-60	18,938 MIPS	2.6 GHz
2007	Intel Core 2 QX9770	59,455 MIPS	3.2 GHz
2011	Intel Core i7 990x	159,000 MIPS	3.46 GHz
2015	Inel Core i7 5960X	238,310 MIPS	3.0 GHz

- how fast is a nanosecond?

Nanoseconds

- speed of light =186,000 miles/sec, $3x10^8$ meters/sec (actually 299,792,458)
- nanosecond = 10^{-9} seconds
- thus 30 centimeters or 11.8 inches

Speed vs. Physics

- the "clock" is like a drummer in the band: the faster it beats, the faster the operations.
- in 1 clock tick (at 1 GHz), electricity can travel 1 ft; at 2GHz, 6 inches; at 4GHz, 3 inches
- it becomes very difficult to get electricity to travel any distance in that short of time!

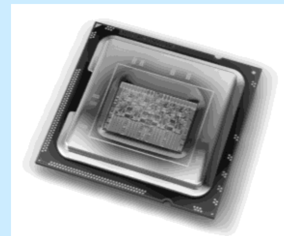
How to Get Around Physics

- so if physics is getting in the way (and it is), you find a way to get around it.
- if you can't make processors faster, what do you do?

 55

Not Faster, but More CPUs

- now more transistors mean more CPUs on a single chip
  - quad-core CPU



 56

Computer Parts

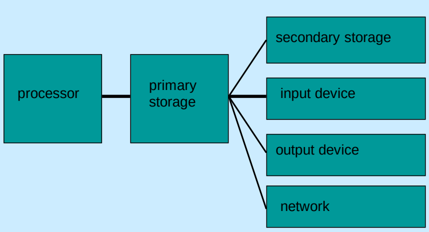
 57

Main Components

- people!!!
- hardware
  - physical devices: processor, memory, keyboard, monitor, mouse, etc.
- software
  - executable programs: word processor, spread sheet, internet browser, etc.

 58

Hardware



 59

Processor

- the processor is the "brain"of a computer
- the processor controls the other devices as well as performing calculations

 60

Primary Storage

- stores instructions and data for current program(s)
- other names: primary or main memory, RAM (Random Access Memory)
- memory is dynamic so it requires power to retain information
- often in terms of gigabytes (billion-bytes)

 61

Peripheral Devices

- secondary storage devices
  - disk (hard & floppy), tape, usb drives, flash drives, etc.
- input devices
  - keyboard, mouse, camera, mic, etc.
- output devices
  - monitor, printer, speaker, etc.
- network
  - wireless, bluetooth, ethernet, etc.

 62

Secondary Storage

- nonvolatile -- information is recorded magnetically so power is not needed
- disks hold terabytes (trillions of bytes)
- cheap, but slow
  - RAM access is a hundred CPU clock ticks
  - disk access is a million CPU clock ticks
- not directly accessed by CPU

 63

Software

- the programs available for execution
- simple classification
  - system software
  - application software

 64

Fetch-Decode-Execute-Store

CPU operates on a simple (but fast) cycle:

- ***Fetch***: fetch instruction from memory
- ***Decode***: decode requirements (args, etc.)
- ***Execute***: perform operation
- ***Store***: move needed results to memory
- repeat

 65

What is a Program?

- a sequence of instructions written in machine language that tells the CPU to take action (e.g., add two numbers) in a specific order
- in this course we will learn to create programs

 66

Program Storage

- machine language instructions are encoded as bit patterns (binary, remember our transistors)
- memory can only hold binary info
- a bit is represented by two-states, e.g. L-R magnetism, high-low voltage
- it takes many bits to represent reasonable amounts of information

 67

Binary Numbers

- the switch nature of transistors make storing numbers in binary a natural fit
- binary is a change of base for our number system, base 2
- in a number, its position represents powers of 2

 68

Example

- decimal uses digits 0-9 and positions in a number as powers of 10
  $735_{10} = 7*10^2 + 3*10^1 + 5*10^0$
- binary users digits 0,1 and positions in a number as powers of 2
  $101_2 = 1*2^2 + 0*2^1 + 1*2^0$
- we can convert back and forth
  $101_2 = 5_{10}$

 69

Binary to Decimal

- rule of thumb: every 10 powers of 2 is about 3 powers of 10
- $2^{10} = 1024 \approx 10^3 = 1000$ (kilobyte)
- $2^{20} = 1{,}048{,}576 \approx 10^6 = 1{,}000{,}000$ (megabyte)
- $2^{30} = 1{,}073{,}741{,}824 \approx 10^9 = 1{,}000{,}000{,}000$ (gigabyte)

 70

Bits, Bytes and Words

- bit: a single 1 or 0
- byte: 8 bits
- word: a computer's standard unit of storage
  - a typical computer is a 32-bit computer, meaning its word size is 32 bits (4 bytes)
  - many processors today are 64-bit

 71

Floating Point

- the approximation of decimal point values such as 1.23, 0.01, 3.14159
- what is the exact value of 1/3 as a decimal point number?
  - an infinite sequence which we approximate when using a decimal number
- remember that floating point is approximate!

 72

Representing Data in Binary

- letters are represented as numbers!
- both ASCII and Unicode are encodings for particular letters
- ASCII (American Standard Code for Information Interchange) is used for English letters
  - superseded by a subset of Unicode called UTF-8 (see appendix E)

 73

UTF-8 Table (First Few Rows)

Char	Dec	Char	Dec	Char	Dec	Char	Dec
NUL	0	SP	32	@	64	`	96
SOH	1	!	33	A	65	a	97
STX	2	"	34	B	66	b	98
ETX	3	#	35	C	67	c	99
EOT	4	$	36	D	68	d	100
ENQ	5	%	37	E	69	e	101
ACK	6	&	38	F	70	f	102
BEL	7	'	39	G	71	g	103
BS	8	(	40	H	72	h	104

 74

Images

- digital images consist of individual colors in a matrix
- each individual color is called a pixel
- the color of a pixel can be encoded using numbers

 75

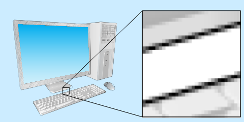


FIGURE 0.14 A computer display picture with a close-up of the individual pixels. [Juliengrondin/Shutterstock]

 76

Sound

- we can sample a sound wave at a very high frequency and measure the height of the sound
- we can represent those measurements as numbers and use them to recreate a sound wave

 77

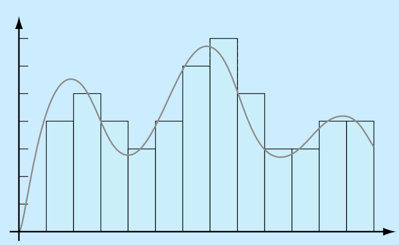


FIGURE 0.15 A sound wave and the samples of that sound wave (blue bar height) over time.

 78

How Many Numbers?

 79

Binary

- byte = 8 bits
  e.g. 10110010
- storage devices come in large quantities
  1 KB (kilobyte) = 2^{10} bytes = 1024 bytes
  1 MB (megabyte) = 2^{20} bytes = 1,048,576 bytes
  1 GB (gigabyte) = 2^{30} bytes =1,073,741,824 bytes

 80

Kilobyte (1000 bytes)

- 2 KB: a typewritten page
- 10 KB: an encyclopedic page
- 50 KB: a compressed document
- 100 KB: a low-resolution photo

 81

Megabyte (1,000,000 bytes)

- 1 MB: a small novel **or** 3.5 inch floppy disk
- 2 MB: a high resolution photograph
- 5 MB: the complete works of Shakespeare
  or 30 seconds of video
- 100 MB: 1 meter of shelved books
- 500 MB: a CD-ROM

 82

Gigabyte (1,000,000,000 bytes)

- 1 GB: a pickup truck filled with paper
  or a symphony
  or a movie
- 20 GB: the works of Beethoven as sound
- 50 GB: a library floor of books
- 100 GB: standard Blu-ray capacity

 83

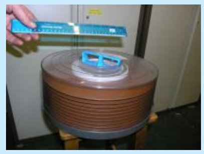
Terabyte (1,000,000,000,000 bytes)

- 1 TB: all X-rays in a large hospital
  or 50000 trees made into paper and printed
  or daily rate of EOS data (1998)
- 2 TB: an academic research library
- 20 TB: photos/month on facebook
- 100 TB: U.S. Library of Congress, all types of data, 2009
- 530 TB, all videos on youtube

 84

Terabyte in Your Hand

- 1 TB = 1000 Gigabytes
  1500 music CD's
  200 DVD movies (16 days worth)
  1/10 of the Library of Congress
  - Western Digital: 1 TB, $269 (2006)
  - today, 10 TB $260
- history
  - first disk: IBM 1956
    - (4 MB, 50 24-inch platters)
  - 1GB in 35 years
  - 500 GB in 14 more years
  - 1000 GB = 1 TB in 2 more years



 85

Petabyte
(1000 TB or 1,000,000 GB)

- EMC Symmetrix DMX-3
  - 1.2 PB = 2400 500GB drives ($4 million 2006)
- Backblaze (build your own)
  - $120,000, 2009
- Kazza shared 54 PB (2005)
- Google cluster has 4 PB of RAM, processes 20 PB a day
- all data recorded in 2003: 2500 PB
- WoW, 1.3 PB to maintain its game

*(1,125,899,906,842,624 bytes = 2^{50})

 86

Petabyte

- **In round numbers, a book is a megabyte. If you read one book a day for every day of your life for 80 years, your personal library will amount to less than 30 gigabytes. Remember a petabyte is 1 million gigabytes so you will still have 999,970 gigabytes left over.**
- **How many pictures can a person look at in a lifetime (4 Mbytes each)? I can only guess, but 100 images a day certainly ought to be enough for a family album. After 80 years, that collection of snapshots would add up to 30 terabytes. So your petabyte disk will have 970,000 gigabytes left after a lifetime of high quality photos and books.**
- **What about music? MP3 audio files run a megabyte a minute, more or less. At that rate, a lifetime of listening – 24 hours a day, 7 days a week for 80 years – would consume 42 terabytes of disk space. So with all your music and pictures and books for a lifetime you will have 928,000 gigabytes free on your disk.**
- **The one kind of content that might possibly overflow a petabyte disk is video. In the format used on DVDs, the data rate is about two gigabytes per hour. Thus the petabyte disk will hold some 500,000 hours worth of movies; if you want to watch them all day and all night without a break for popcorn, they will actually fill up your petabyte drive if you have a lifetime of video on it as it will give you 57 years of video**

 87

Exabyte
(1,000,000,000,000,000,000 bytes)

- 5 exabytes: all words ever spoken (as text) by human beings
- 5-8 EB: global internet traffic monthly
- 500 EB: total world digital content (2009)

 88

Zettabyte (10^{21})
1,000,000,000,000,000,000,000

- audio of everything ever spoken, by anyone, at anytime in history, 16KHz, 16 bit audio: 42 ZB

 89

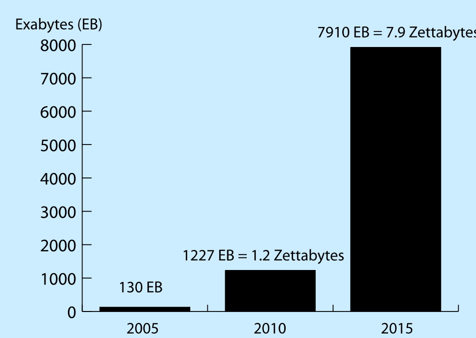


FIGURE 0.16 Estimates of data created and stored. (IDC Digital Universe)

 90

The Rules

1. Think before you program.
2. A program is a human-readable essay on problem solving that also executes on a computer.

 91