

Chapter 3 Software
3.1 - Language of Computers

Grading:

Notes: _____/20
Task 1: _____/10
Task 3: _____/5
Task 4: _____/5
Lesson Review: _____/20
Computer Language: _____/20
Your Name: _____/10
Inside & Out Bytes _____/10
Total points: _____/100

Essential Question

- How is a strong understanding of math important to computer programming?

Section 3.1 Learning Goals

After completing this section, you will be able to:

- Compare mechanical and electronic computers.
- Explain various number systems.
- Discuss computer programming languages.

Competencies

- 6670.37 Explore basic binary concepts and their applications in the information technology world.
- 6670.62 Identify basic problems associated with computer hardware, operating system, and application software

Terms

- American Standard Code for Information Interchange (ASCII)
- assembly language
- bit
- byte
- bytecode
- code
- compiler
- computer algorithm
- data type
-
- encoding
- high-level programming language
- interpreter
- low-level programming language
- machine language
- object-oriented languages
- procedural languages
- programs
- Unicode

Electronic Computers

- Electronic computers have no moving parts
- All _____ represented as
 - On/off
 - 1/0
 - True/false
- Files are patterns of 1s and 0s
- *Binary* _____, or **bit**, is the basic building block for communication in an electronic computer
- Eight bits equal _____ **byte**

Number Systems

- _____ Numbers
 - Positional system
 - Based on place value and base 10
 - Digits used are 1, 2, 3, 4, 5, 6, 7, 8, 9, 0
 - Places represent powers of 10
- _____ Numbers
 - Positional system
 - Based on place value and base 2
 - Digits used are 1 and 0
 - Places represent powers of 2
- _____ Numbers
 - Positional system
 - Based on place value and base 16
 - Digits used are 0–9, A, B, C, D, E, F
 - Places represent powers of 16

Binary Code

0100001001101001011011100110000101110010011110010010000001000011011011110110
010001100101

OBJECTIVES

Computers are not as smart as you think, in fact, they only know 2 things... 0 and 1

Everything else is told to the computer by a _____, this is called programming.

Computers have millions of tiny _____ inside

For a computer

0 means turn a circuit _____.

1 means turn a circuit _____.

Humans need to learn the computers language to be able to _____ computers what to do.

The computers language is _____.

Here is an example of what binary code looks like:

0101010001101000011010010111001100100000011010010111001100100000011001010110
000101110011011110010000110100001010

The computer can only use 0s and 1s

So, for a computer numbers go like is this...

Denary (our numbers)	Binary
0	0
1	1
2	10
3	11
4	100
5	101

6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100

How am I supposed to remember that?!

128	64	32	16	8	4	2	1

To make the number 5, place a 1 under the numbers that you need and a 0 under the numbers that you don't...

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

So, 5 for us, is 00000101 for the computer

We can forget the first few 0s and our binary number 5 looks like this...**101**

TASK 1: Now use your table to convert these numbers to binary:

Decimal #	Binary	Decimal #	Binary
3		20	
2		42	
30		47	
6		71	
10		146	
17		254	

Is there any number that CAN'T be made?

Now it's time to convert binary to OUR digits

128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

128	64	32	16	8	4	2	1
0	0	1	0	0	1	0	1

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	1

TASK 2: Can you convert these binary numbers into denary?

Binary	Decimal #	Binary	Decimal #
00011000		10100101	
00000101		10111100	
00010110		11010111	
01101011		11100110	
01111000		11111010	
10100100		00000001	

TASK 3: Work with a partner and see if you can de-code these facts...

- More than 110010% of the people in the world have never made or received a telephone call.
- Gorillas sleep as much as 1110 hours per day.
- You can tell the sex of a horse by its teeth. Most males have 101000, females have 100100.
- On average, 1100 new born will be given to the wrong parents daily.

- A giraffe can clean its ears with its 10101-inch tongue!

- A man named Charles Osborne had the hiccups for 1000101 years!

Binary code isn't just about numbers. Every number, letter and symbol on your keyboard needs to be converted into binary so the computer can understand it.

We can see all these conversions in something called an ASCII table.

From the ASCII table...

Symbol	Decimal	Binary	HEX
A	65	01000001	41
B	66	01000010	42
C	67	01000011	43
D	68	01000100	44
E	69	01000101	45
F	70	01000110	46
G	71	01000111	47
H	72	01001000	48
I	73	01001001	49
J	74	01001010	4A
K	75	01001011	4B
L	76	01001100	4C
M	77	01001101	4D

N	78	01001110	4E
O	79	01001111	4F
P	80	01010000	50
Q	81	01010001	51
R	82	01010010	52
S	83	01010011	53
T	84	01010100	54
U	85	01010101	55
V	86	01010110	56
W	87	01010111	57
X	88	01011000	58
Y	89	01011001	59
Z	90	01011010	5A

Symbol	Decimal	Binary	HEX
a	97	01100001	61
b	98	01100010	62
c	99	01100011	63
d	100	01100100	64
e	101	01100101	65
f	102	01100110	66
g	103	01100111	67
h	104	01101000	68
i	105	01101001	69
j	106	01101010	6A
k	107	01101011	6B
l	108	01101100	6C
m	109	01101101	6D

n	110	01101110	6E
o	111	01101111	6F
p	112	01110000	70
q	113	01110001	71
r	114	01110010	72
s	115	01110011	73
t	116	01110100	74
u	117	01110101	75
v	118	01110110	76
w	119	01110111	77
x	120	01111000	78
y	121	01111001	79
z	122	01111010	7A

Every time you type a letter on your keyboard, the computer understands it as loads of 0s and 1s!

Here are some other characters you might need from the ASCII table...

Symbol	Decimal	Binary	HEX
Space	32	00100000	20
!	33	00100001	21
"	34	00100010	22
#	35	00100011	23
\$	36	00100100	24
%	37	00100101	25
&	38	00100110	26
'	39	00100111	27
(40	00101000	28
)	41	00101001	29
*	42	00101010	2A

+	43	00101011	2B
,	44	00101100	2C
-	45	00101101	2D
.	46	00101110	2E
/	47	00101111	2F
:	58	00111010	30
;	59	00111011	31
<	60	00111100	32
=	61	00111101	33
>	62	00111110	34
?	63	00111111	35

Symbol	Decimal	Binary	HEX
@	64	01000000	40
[91	01011011	5B
\	92	01011100	5C
]	93	01011101	5D
^	94	01011110	5E

_	95	01011111	5F
`	96	01100000	60
{	123	01111011	61
	124	01111100	62
}	125	01111101	63
~	126	01111110	64

Binary Game: <https://studio.code.org/projects/applab/iukLbcDnzqgoxuu810unLw>

Code

- Encoding Instructions
- **Encoding** is the process of _____ human-readable data and computer programs into a computer-readable format
- **Code** is the result of the encoding process
- Encoding Data
 - **American Standard Code for Information Interchange (_____)** is a standard for representing text that most computers support
 - **Unicode** is a system for _____ text characters in which two bytes are assigned to each character
- Computer Programming Languages
 - _____-Level Languages
 - Contain instructions that are far removed from the instruction the computer CPU uses
 - Programming software contains a _____, which converts the programmer's code into code the CPU can understand
 - **Interpreter** converts instructions to code the _____ can understand as the program is executing
 - **Bytecode** is a set of instructions composed of _____ numeric codes, constants, and references that can be efficiently processed by an interpreter
 - **Procedural languages** are computer programming languages in which instructions are gathered into _____ called procedures
 - **Object-oriented languages** contain data structures and _____ that can be performed on those structures
 - **Low-Level Languages**
 - Very _____ to the instruction set used by the CPU
 - **Machine language** is a _____-level language composed of the 0s and 1s the CPU uses
 - **Assembly language** is very close to machine language, but the CPU cannot directly understand it
 - _____ for a person to read and write
- Data Types Used in Computer Programming
 - **Data type** is the description of _____ or information that can be accepted
 - Stored in different ways
- Algorithms
 - **Computer algorithm** is a series of _____ used to perform an action
 - Map of what needs to be done
 - Programmers write the code to activate the algorithm
 - Types of algorithms
 - _____ algorithms are processed once and solutions are output
 - _____ algorithms are repeated until a condition is met

Section 3.1 Review

Hands-on Example 3.1.2 - Encode the text string *Hello World!* Using **HEX** notation. Use the ASCII chart shown in Figure 3-7 or pages 6-8 in the notes to locate the correct symbols.

1. Locate the codes for the characters in the word *Hello*.
2. Locate the code for a space.
3. Locate the codes for the characters in *World*.
4. Locate the code for the exclamation point.
5. Write the codes (those you did in #1-4) in order from left to right to form the code for the entire text string. Use the 0x prefix.

6. Convert this hex code into its binary equivalent: 0xF6A9 (0x tells the computer it is HEX. So started with the 'F')

7. Convert this binary code into its hex equivalent: 1101101010011 (*hint work from the right to left, break into groups of 4*)

8. What is the name of a program that converts higher-level programming language into machine language?

Decimal	Binary	Hex
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	1 0000	10

9. What is the difference between a compiler and an interpreter?

10. What is the basic difference between a linear algorithm and an iterative algorithm?

11. A memory address is stated as 0x14DE.
What is the binary equivalent? (*Hint work from the right to left*)

12. Programs written in a high-level language are always:
 - a. interpreted at run-time.
 - b. readable by the CPU.
 - c. translated into machine language.
 - d. linked for all computers.

13. The value 3.1415926 would belong to which data type? (Might want to look at chapter 2)
 - a. integer
 - b. floating point
 - c. string
 - d. date

14. Languages are used in computer programming to:
 - a. speaks with a computer.
 - b. write computer instructions at a human level.
 - c. segment the hard drive.
 - d. communicates with the hardware.

Application and Extension of Knowledge

Computer Languages: Identify five computer programming languages. Classify each as a high- or low-level language. Prepare a Google Slide presentation explaining why you classified each as you did. Include:

- Title slide including your name
- The language
- High or low-level language
- Relevant picture
- Why you classified each language

- Share with teacher when done
- You can do one slide per language

NOTICE THERE IS MORE ON THE NEXT PAGE

Your Name

What is the Binary equivalent of your name?

Computers: Inside & Out – Bits & Bytes

Bytes Worksheet

Each rectangle below represents one byte made up of 8 bits (small rectangles). Label the byte by assigning 1 to each on bit, and 0 to off bit. On bits are white and off bits are black.

1.	Black	White	Black	White	White	White	White	Black

2.	White	Black	Black	White	White	Black	White	White

3.	White	White	White	Black	White	Black	Black	Black

4.	Black	Black	White	White	Black	White	White	White

5.	White	Black	White	White	Black	White	White	White

6.	White	Black	White	Black	White	Black	White	Black

7.	Black	Black	White	Black	White	Black	Black	White

8.	White	Black	Black	White	White	White	Black	Black

9.	White	Black	White	Black	White	Black	Black	Black

10.	Black	White	Black	Black	Black	White	White	White

11.	White	Black	White	Black	Black	White	Black	White

12.	White	Black	White	Black	Black	Black	White	Black