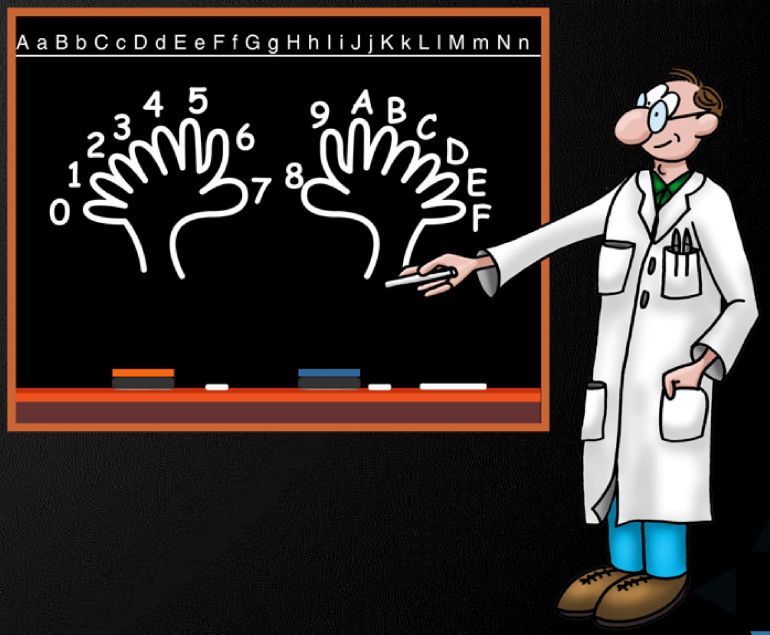
Telerik Academy

Numeral Systems

Binary, Decimal and Hexadecimal Numbers



C# Advanced

Telerik Software Academy

https://telerikacademy.com

Telerik Academy

Table of Contents

Telerik Academy

Numeral Systems

Telerik Academy

Decimal Numbers

- Decimal numbers (base 10)
  - Represented using 10 numerals: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Each position represents a power of 10:

$$401 = 4*10^2 + 0*10^1 + 1*10^0 = 400 + 1$$

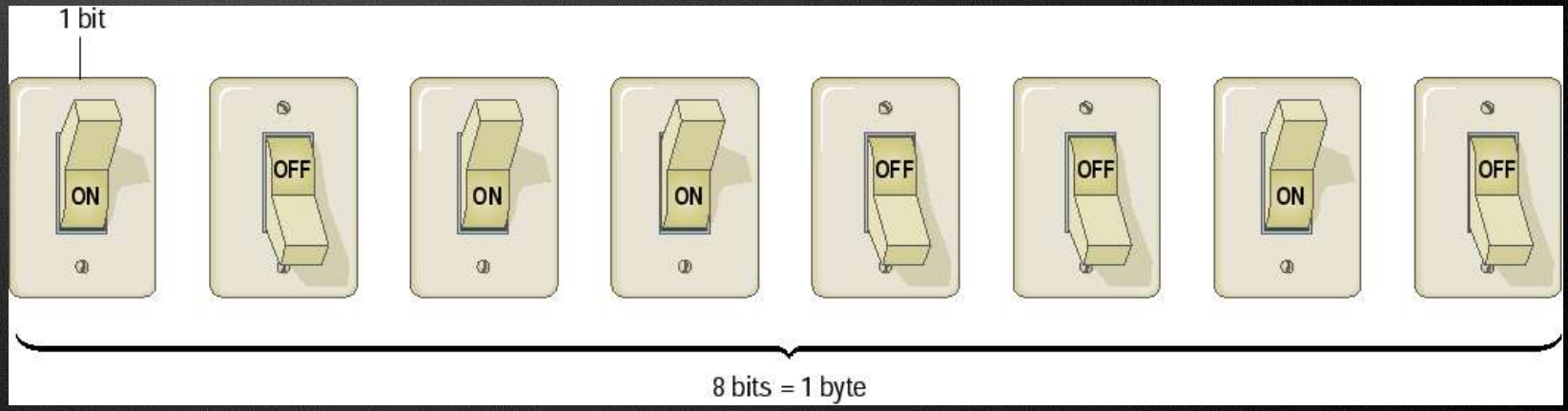
$$130 = 1*10^2 + 3*10^1 + 0*10^0 = 100 + 30$$

$$9786 = 9*10^3 + 7*10^2 + 8*10^1 + 6*10^0 = 9*1000 + 7*100 + 8*10 + 6*1$$

Telerik Academy

Binary Numeral System

- Binary numbers are represented by sequence of bits (smallest unit of information – 0 or 1)
  - Bits are easy to represent in electronics



Telerik Academy

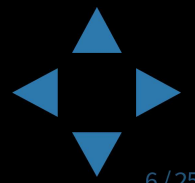
Binary Numbers

- Binary numbers (base 2)
  - Represented by 2 numerals: 0 and 1
- Each position represents a power of 2:

$$101_b = 1*2^2 + 0*2^1 + 1*2^0 = 100_b + 1_b = 4 + 1 = 5$$

$$110_b = 1*2^2 + 1*2^1 + 0*2^0 = 100_b + 10_b = 4 + 2 = 6$$

$$110101_b = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 32 + 16 + 4 + 1 = 53$$



Telerik Academy

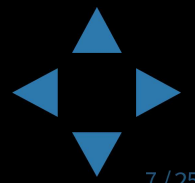
Binary to Decimal Conversion

- Multiply each numeral by its exponent:

$1001_b = 1*2^3 + 1*2^0 = 1*8 + 1*1 =$
$= 9$

$0111_b = 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0 =$
$= 100_b + 10_b + 1_b = 4 + 2 + 1 =$
$= 7$

$110110_b = 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 =$
$= 100000_b + 10000_b + 100_b + 10_b =$
$= 32 + 16 + 4 + 2 =$
$= 54$

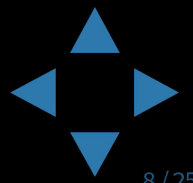


Telerik Academy

Decimal to Binary Conversion

- Divide by 2 and append the reminders in reversed order:
  - 500/2 = 250 (0)
  - 250/2 = 125 (0)
  - 125/2 = 62 (1)
  - 62/2 = 31 (0)
  - 31/2 = 15 (1)
  - 15/2 = 7 (1)
  - 7/2 = 3 (1)
  - 3/2 = 1 (1)
  - 1/2 = 0 (1)

$500_d = 111110100_b$

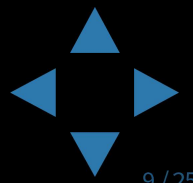


Telerik Academy

Hexadecimal Numbers

- Hexadecimal numbers (base 16)
  - Represented using 16 numerals: `0`, `1`, ... `9`, A, B, C, D, E and F
  - Usually prefixed with `0x`
- Hex digits value:

0 - 3	4 - 7	8 - 11	12 - 15
0 → 0x0	4 → 0x4	8 → 0x8	12 → 0xC
1 → 0x1	5 → 0x5	9 → 0x9	13 → 0xD
2 → 0x2	6 → 0x6	10 → 0xA	14 → 0xE
3 → 0x3	7 → 0x7	11 → 0xB	15 → 0xF



Telerik Academy

Hexadecimal Numbers

- Each position represents a power of 16:

$$9786_{hex} = 9*16^3 + 7*16^2 + 8*16^1 + 6*16^0 =$$
$$= 9*4096 + 7*256 + 8*16 + 6*1 =$$
$$= 38790$$

$$0xABCDEF_{hex} = 10*16^5 + 11*16^4 + 12*16^3 +$$
$$13*16^2 + 14*16^1 + 15*16^0 =$$
$$= 11259375$$

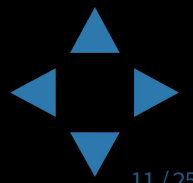
Telerik Academy

Hexadecimal to Decimal Conversion

- Multiply each digit by its exponent

$$1F4_{hex} = 1*16^2 + 15*16^1 + 4*16^0 =$$
$$= 1*256 + 15*16 + 4*1 =$$
$$= 500_d$$

$$FF_{hex} = 15*16^1 + 15*16^0 =$$
$$= 240 + 15 =$$
$$= 255_d$$

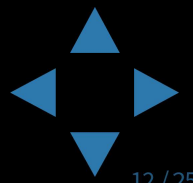


Telerik Academy

Decimal to Hexadecimal Conversion

- Divide by 16 and append the reminders in reversed order
  - 500/16 = 31 (4)
  - 31/16 = 1 (F)
  - 1/16 = 0 (1)

$500_{d} = 1F4_{hex}$



Telerik Academy

Binary to Hexadecimal (and reverse) Conversion

- The conversion from binary to hexadecimal (and back) is straightforward: each hex digit corresponds to a sequence of 4 binary digits:

0 - 3	4 - 7	8 - B	C - F
0x0 = 0000	0x4 = 0100	0x8 = 1000	0xC = 1100
0x1 = 0001	0x5 = 0101	0x9 = 1001	0xD = 1101
0x2 = 0010	0x6 = 0110	0xA = 1010	0xE = 1110
0x3 = 0011	0x7 = 0111	0xB = 1011	0xF = 1111

Telerik Academy

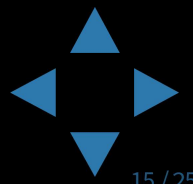
Numbers Representation

Positive and Negative Integers and Floating-Point Numbers

Telerik Academy

Representation of Integers

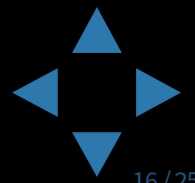
- A `short` is represented by 16 bits
  - $100 = 2^6 + 2^5 + 2^2 = 00000000\ 01100100$
- An `int` is represented by 32 bits
  - $65545 = 2^{16} + 2^3 + 2^0 = 00000000\ 00000001\ 00000000\ 00001001$
- A `char` is represented by 16 bits
  - $'0' = 48 = 2^5 + 2^4 = = 00000000\ 00110000$



Telerik Academy

Positive and Negative Numbers

- A number's sign is determined by the Most Significant Bit (MSB)
  - Only in signed integers: `sbyte`, `short`, `int`, `long`
  - Leading `0` means positive number
    - Leading `1` means negative number
  - *Example*: (8 bit numbers)
    - $0XXXXXXX_b$ > 0 e.g. 00010010b = 18
    - 00000000_b = 0
    - $1XXXXXXX_b$ < 0 e.g. 10010010b = -110



Telerik Academy

Positive and Negative Numbers

- The largest positive 8-bit `sbyte` number:
  - $127 = (2^7 - 1) = 01111111_b$
- The smallest negative 8-bit number:
  - $-128 = (-2^7) = 10000000_b$
- The largest positive 32-bit `int` number:
  - $2\,147\,483\,647\ (2^{31} - 1) = 01111\ldots11111_b$
- The smallest negative 32-bit number:
  - $-2\,147\,483\,648\ (-2^{31}) = 10000\ldots00000_b$

Representation of 8-bit Numbers

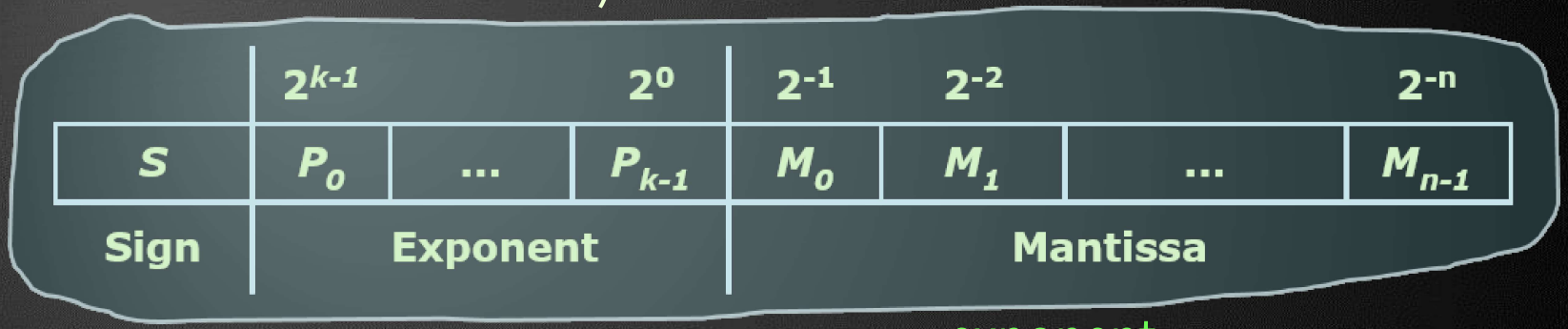
- Positive 8-bit numbers have the format 0 XXXXXXX
  - Their value is the decimal of their last 7 bits (XXXXXXX)
- Negative 8-bit numbers have the format 1 YYYYYYY
  - Their value is 128 (2^7) minus (-) the decimal of YYYYYYY
  - $10010010_b = 2^7 - 10010_b =$ 128 - 18 = -110

```
+127 = 01111111
...
+3   = 00000011
+2   = 00000010
+1   = 00000001
+0   = 00000000
-1   = 11111111
-2   = 11111110
-3   = 11111101
...
-127 = 10000001
-128 = 10000000
```

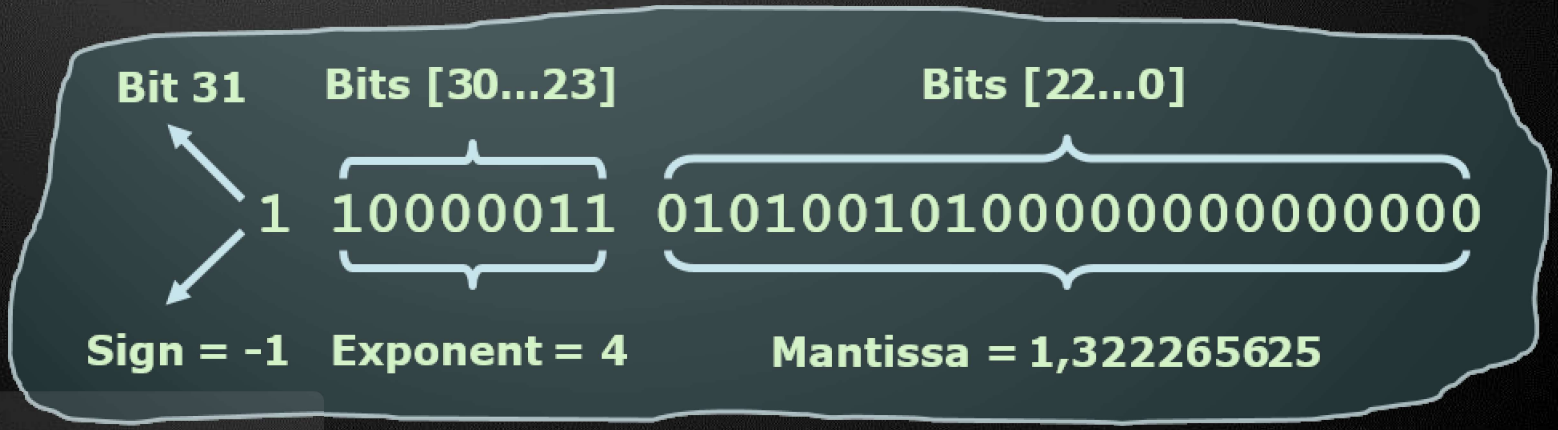
Telerik Academy

Floating-Point Numbers

- Floating-point numbers representation (according to the IEEE 754 standard*):



- Example: -21.15625 -> sign * $2^{exponent}$ * mantissa



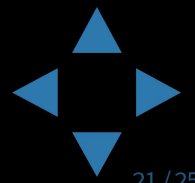
Telerik Academy

Text Representation in Computer Systems

Telerik Academy

How Computers Represent Text Data?

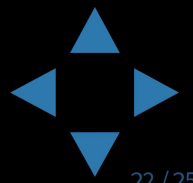
- `Text encoding` is a system that uses binary numbers (1 and 0) to represent characters
  - Letters, numerals, symbols etc.
- In the `ASCII` encoding each character consists of 8 bits (one byte) of data
  - ASCII is used in nearly all personal computers
- In the `Unicode` (`UTF-16`) encoding each character consists of 16 bits (two bytes)
  - Can represent many alphabets



Telerik Academy

Character Codes – ASCII Table

Binary code	Decimal code	Character
01000001	65	A
01000010	66	B
01000011	67	C
01000100	68	D
00100011	35	#
01100000	48	0
00110001	49	1
01111110	126	~



Telerik Academy

Strings of Characters

- Strings are sequences of characters
  - Null-terminated (like in C)

- Represented by array

- Characters in the strings can be:
  - `8 bit` (`ASCII` / `windows-1251` / ...)
  - `16 bit` (`UTF-16`)

Telerik Academy

C# Numeral Systems

Questions?

Telerik Academy

Free Training @ Telerik Academy

- Fundamentals of C# Programming Track of Courses
  - csharpadvanced
    - Telerik Software Academy
  - telerikacademy.com
    - Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy
    - Telerik Academy Learning System
  - telerikacademy.com