1. Log in
2. Open Netbeans and Moodle
3. Did you finish Unit 1 WS 03?
4. Did you enter your answers online?

Sep 28-7:21 AM

Number variables thus far...

int double

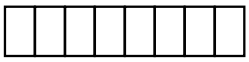
There is actually another called ...

byte

You will not be tested on byte, but ... we need to study it to help us begin to understand some concepts.

Sep 28-9:28 AM

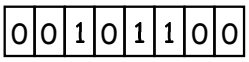
Data type 'byte' only uses 8-bits of memory ...



* think of each box above as one bit of memory
* each bit can only store 0 or 1 ... it is "ON" or "OFF"

Oct 2-12:28 PM

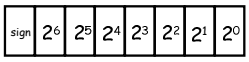
Data type 'byte' only uses 8-bits of memory ...



Binary code ... woohoo!

Oct 2-12:28 PM

Data type 'byte' only uses 8-bits of memory ...



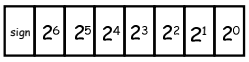
0 0 1 0 1 1 0 0 Is it used???

p 0+ 2^5+0+ 2^3+2^2+0+ 0

pos 32 + 8 + 4 = 44

Oct 2-12:28 PM

So the LARGEST byte number possible is ...



0 1 1 1 1 1 1 Is it used???

64+32+16+8+4+2 + 1 = 127

Amazingly, this is 2^7-1 7 bits of memory

Oct 2-12:28 PM

Ignoring the sign-bit, here is binary code for the first 15 numbers

Try them ...

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

Table taken from the following website:

http://www.binaryhexconverter.com/binary-to-decimal-converter

Do you want more practice? Write your own binary code, try to turn it into a decimal number, and use this website to check your work!

Oct 2-12:53 PM

Try some binary (ignore the sign bit) ...

0 1 1 1 0 1

Remember: Start on the right side

1 0 1 1 0 0 0

Oct 2-1:54 PM

So, now the big MUST KNOW concept ...

int variables use 32-bits of memory

reserved for sign

* So there are 31 bits remaining for information
* Largest possible integer you can make is $2^{31}-1$
* The minus 1 is because we need to store zero.
* The value $2^{31}-1$ is stored in JAVA as Integer.MAX_VALUE

Oct 2-12:28 PM

AP Says you must know ...

Integer.MAX_VALUE = $2^{31}-1$

Integer.MIN_VALUE = -2^{31}

You can access these in your code!

*** MIN_VALUE does not have the minus 1 since we don't need to store zero. We start at -1. You won't be tested on negatives.

Oct 2-1:48 PM

Now ... how do you go from decimal to binary?

1. Divide by 2 and write the remainder
2. Continue this process until after you get a division answer of zero with a remainder of 1
3. Read your binary code "bottom-up"
4. Verify you have 8, 16, 24, or 32 sets of digits (memory is allocated in 8-bit sets)

Let's take a look at this process ...

Oct 2-1:54 PM

Now ... how do you go from decimal to binary?

1. Divide by 2 and write the remainder
2. Continue until after you do 1 ÷ 2 = 0 R
3. Read your binary code "bottom-up"
4. Verify you have 8, 16, 24, or 32 digits (memory is allocated in 8-bit sets)

What is 188 as binary code?

188 ÷ 2	Remainder
94	0
47	0
23	1
11	1
5	1
2	1
1	0
0	1

Binary: 10111100

Oct 2-1:54 PM

Now ... how do you go from decimal to binary?

What is 861 as binary code?

1. Divide by 2 and write the remainder
2. Continue until after you do 1 ÷ 2 = 0 R1
3. Read your binary code "bottom-up".
4. Verify you have 8, 16, 24, or 32 digits (memory is allocated in 8-bit sets)

861 ÷2	Remainder
430	1
215	0
107	1
53	1
26	1
13	0
6	1
3	0
1	1
0	1

Binary: 00000011 01011101

Oct 2-1:54 PM

Checklist for today ...

- ☐ Understand today's lesson
- ☐ Complete Unit 1 Worksheet 04
- ☐ Enter Worksheet Answers in Moodle (counts as a grade, retake once)
- ☐ Review documents for converting can be found on course website

Sep 28-9:44 AM