Java Programming
Hexadecimal Numbers

You've learned about binary numbers (base 2) and decimal numbers (base 10). Programmers usually use hexadecimal numbers (base 16) because it's more convenient. Here's why:

Each **bit** of data is a 0 or a 1. If we line up 8 bits of data together, we get this:

The biggest positive number we can hold in 8 bits of data is by putting a **1** in each slot:

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

In decimal, $11111111_2 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255_{10}$

To make it easier to talk about binary, programmers (and computers) group 8 bits together into 1 "byte" and talk about each byte by using 2 numbers.

8 bits == 1 byte

To talk about 1 byte, programmers talk about the "low order bits" and the "high order bits". In the following diagram, the high order bits are shaded.

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

Programmers take the 4 high order bits and talk about them as one number. Same for the low order bits:

High bits			
8	4	2	1
1	**0**	**0**	**1**

Low bits			
8	4	2	1
0	**1**	**1**	**1**

Then you can talk about one byte using **2 digits**: one digit for the high bits and one digit for the low bits. The byte above holds the value "97" in decimal (9 in the high bits and 7 in the low bits).

Weird-looking numbers such as "6F"

What happens when all 4 of the low or high order bits are full, like this?

High bits			
8	4	2	1
1	**1**	**1**	**1**

Low bits			
8	4	2	1
1	**1**	**1**	**1**

Contact: greenv@portlandschools.org

Uh oh! Now the high bits and low bits are each equal to "15" in decimal, but we can't say "1515" to describe them, because that's more than 2 digits! To get around this problem, we use **base 16**, or **hexadecimal** numbers. In base 16, the digits go from 0 to 9 and then from A to F. "A" is the same as "10" in decimal. "F" is the same as "15" in decimal.

The byte above would be represented as "FF" in hexadecimal. All slots are full, so each 4 bits are worth 15, or "F" in hex.

Practice it Yourself
All questions are worth 5 points each.

Give the 2-digit hexadecimal value of the following bytes.

1. 11111111 = ____ ____

2. 10101110 = ____ ____

3. 11101011 = ____ ____

Write the byte described by each hexadecimal number by filling in a 0 or a 1 for each bit.

4. 5F

5. 6C

6. DD

7. 58
