

Program #1 - 1101 1100

This is a short Java exercise on binary number formats.

- Due: **Fri Jan 17, 2013**
- Worth: 5 points

Good luck!

1. Description

Write a Java program that reads an integer and prints its binary representation in a variety of formats that we learned in Appendix A: unsigned binary, unsigned hex, signed-magnitude, 1's complement and 2's complement.

Program 1 is a console program (keyboard input/text output) with no command-line parameters. The user flow is:

1. Ask user the number of bytes in a word
2. Ask user for a number for a number to encode. On a bad number, exit.
3. Print the binary encodings for the number
4. Go to step 2

There's a sample session at the end of this document. More helpful, I'll post a gaggle of test cases on our k: drive, along with output from my solution.

2. Grading

Create a **program1** folder in your k: drive space. I'll look for there for the following:

- Your **README.txt** file where you describe the state of your program.
- Your NetBeans project and Java source code
- Your output files, one per input test file.

3. Notes

I won't nitpick on your error messages but don't crash on bad input data.

Some Java things I used:

- **Scanner** class to get integers from the user. Example: `Scanner.nextInt()`.
- **Modulus operator** to return the remainder. Example: `remainder = num % 2`.
- **Left shift operator** to calculate 2^N . Example: `largest = 1 << numBits`.
- **Right shift operator** to halve a number. Example: `weight = weight >> 1`.

Don't use the byte data type in Java. We want to show the algorithms we use to do binary encodings. I created a simplistic BitString class. It's one field is an array of char's. It has methods like: BitString(numBytes), clear(), invert(), encodeUnsigned(num), setBit(pos, char).

Here is a sample session. User input is in **bold**.

```
> program1
Proram #1 by Bill Krieger

Enter number of bytes: 2

Enter number: 17

Input number=17
  Unsigned binary=0001 0001      (0x11)
  Signed-magnitude=0001 0001
  One's complement=0001 0001
  Two's complement=0001 0001
  Excess 128=1001 0001

Enter number: -60

Input number=-60
  Unsigned binary=undefined
  Signed-magnitude=1011 1100
  One's complement=1100 0011
  Two's complement=1100 0100
  Excess 128=0100 0100

Enter input number: quit
Done. Thank you.
>
```

4. Test files

Please run these small test suites on your program. Here are the three test suites and a link to my results:

1. 0, 17, 100, 127, 128, 220... [test1.txt](#)
2. 0, -1, -17, -100, -127, -128, -129... [test2.txt](#)
3. 4095, 4096, -4095, 60000, -30000, 75000, 220, -220... [test3.txt](#)