**CS201 Lab 5**

**35 points**  **Due**: 02/23/15

**Problem:** We just learned how to convert between binary and decimal numbers. Wouldn’t it be great to have your own program to use to test your results? Let’s make one! But let’s not trust our users to give us exactly what we ask them to input.

**Purpose:** This lab gives you practice with:

* Conversion between binary and decimal numbers
* Protecting your program from bad input using repetition
* Using repetition in your code (loops)

**Details:**

You ask the user to give you an 8-bit unsigned binary number so that you can convert it to decimal (base 10). However, what if the user accidentally only gives you 7 bits? Your program might crash! You therefore need to also **protect your program** by checking if the user has given you exactly 8 bits, and if they haven’t done so you should keep asking them until you have 8 bits. (How do you check if a String is 8 characters long?)

*HINT:* When converting from binary to decimal, you may find it helpful to read the binary number in as a String and then use the .charAt( ) method to get an individual character from the String (e.g., mystring.charAt(0) gets the first character in the String called mystring). Then you can convert that character to a number.

*HINT*: If you have a char variable holding the value ‘1’ but you want the numeric value 1, you can do:

char a = ‘1’; //for example’s sake

int value = Character.getNumericValue(a); //converts it to 1, just like Integer.parseInt does with Strings

**Steps:**

1. Read through the requirements in the Details section and make sure you understand what you need to do.
2. You have a start of an algorithm in your Lab5 repository folder. Make sure you understand the parts of the algorithm already provided, particularly the part that is converting binary to decimal. Work it out on paper on a test number like 00001111 to see if the algorithm works. It is really just following the process we used in class.
3. Complete the algorithm so that it includes the ability to protect the program as specified agove, and **get the professor’s approval before starting to code**.
4. Create a new Java file and save it to your Lab5 repository folder.
5. Write your Java code following your algorithm.
6. Write comments in your code to make it clear what it is doing.
7. Draw a flowchart of your code
8. Create a list of 5 test cases. We haven’t talked about how to get these from the flowchart, so you don’t need to get them from control paths. Think about special cases in the input.
9. Test your program and fix any errors.
10. Include an updated version of the header comments. Many lines should change!

**Extra Credit:**

Once you have the lab completely working, add code to protect the program in case something bad happens. In this case, you should keep asking the user for a binary number until it is only 8 bits long. Another longer or shorter should not be used in the calculation, but should instead ask the user to enter a binary number again.

**Submit:**

1. To GitHub:
   * Your algorithm in the provided Word document
   * Your .java file
   * Your test cases
2. On paper in class:
   * A short reflection about what you learned in lab, what it was like working with your partner, and what gave you the most trouble. (1 per person)
   * A hardcopy of your flowchart (1 copy per pair)