CS2110 Spring 2012
Homework 2

Version: 1

This assignment is due by:

Day: January 24th 2012
Time: 11:54:59pm

Rules and Regulations

Academic Misconduct

Academic misconduct is taken very seriously in this class. Homework assignments are collaborative. However, each of these assignments should be coded by you and only you. This means you may not copy code from your peers, someone who has already taken this course, or from the Internet. You may work with others **who are enrolled in the course,** but each student should be turning in their own version of the assignment. Be very careful when supplying your work to a classmate that promises just to look at it. If he/she turns it in as his own you will both be charged.

We will be using automated code analysis and comparison tools to enforce these rules. **If you are caught you will receive a zero and will be reported to Dean of Students.**

Submission Guidelines

1. You are responsible for turning in assignments on time. This includes allowing for unforeseen circumstances. If you have an emergency let us know ***IN ADVANCE*** of the due time supplying documentation (i.e. note from the dean, doctor's note, etc). Extensions will only be granted to those who contact us in advance of the deadline and no extensions will be made after the due date.

2. You are also responsible for ensuring that what you turned in is what you meant to turn in. No excuses, what you turn in is what we grade. In addition, your assignment must be turned in via T-Square. When you submit the assignment you should get an email from T-Square telling you that you submitted the assignment. If you do not get this email that means that you did not complete the submission process correctly. Under no circumstances whatsoever we will accept any email submission of an assignment. Note: if you were granted an extension you will still turn in the assignment over T-Square.

3. There is a random grace period added to all assignments and the TA who posts the assignment

determines it. The grace period will last **at least one hour** and may be up to 6 hours and can end on a 5 minute interval; therefore, you are guaranteed to be able to submit your assignment before 12:55AM and you may have up to 5:55AM. As stated it can end on a 5 minute interval so valid ending times are 1AM, 1:05AM, 1:10AM, etc. **Do not ask us what the grace period is we will not tell you**. *So what you should take from this is not to start assignments on the last day and depend on this grace period past 12:55AM.* There is also no late penalty for submitting within the grace period. If you can not submit your assignment on T-Square due to the grace period ending then you will receive a zero, no exceptions.

General Rules

1. In addition any code you write (if any) must be clearly commented and the comments must be meaningful. You should comment your code in terms of the algorithm you are implementing we all know what the line of code does.
2. Although you may ask TAs for clarification, you are ultimately responsible for what you submit.
3. Please read the assignment in its entirety before asking questions.
4. Please start assignments early, and ask for help early. Do not email us the night the assignment is due with questions.

Submission Conventions

1. All files you submit for assignments in this course should have your name at the top of the file as a comment for any source code file, and somewhere in the file, near the top, for other files.
2. When preparing your submission you may either submit the files individually to T-Square (preferred) or you may submit an archive (zip or tar.gz only please) of the files.
3. If you choose to submit an archive please don't zip up a folder with the files, only submit an archive of the files we want.
4. Do not submit compiled files that is .class files for Java code and .o files for C code.
5. Do not submit links to files. We will not grade assignments submitted this way as it is easy to change the files after the submission period ends.

Objectives

1. To understand the bitwise operators
2. To use bitwise operators to complete standard tasks

Overview

This assignment consists of two parts, both of which consist of problems related to data types binary numbers and bitwise operations. As always start early or you may run into problems.

(1) A set of questions, which can be found under the 'Tests and Quizzes' tab on T-Square.
(2) Two java files with method headers which you are to complete.

You should complete *both sections* of this homework. Only completing one part will result in a heavy penalty!

Resources

Begin by reading Chapter 2 of your text ("Introduction to Computing Systems"). Already read it? Read it again. No really, reading and understanding this chapter will prove to be of a great advantage on this homework and understanding several of the topics we will cover later in the course.

If you need help with the IEEE floating point standard, the included file 'floating.txt' is a helpful tutorial.

Part 1

Note this part of the assignment is due strictly at 12:55AM Wednesday January 25. There is no additional time with this as we can't set a grace period with a quiz. The grace period only applies to the coding part.

The first part of this assignment is a set of questions that test your ability to work with binary numbers in different representations. You can find these problems under 'Tests and Quizzes' on T-Square.

You can answer the problems in any order, and submit the problems as many times as you like before the submission deadline. Your last submission will be the one that gets graded. Note that when you submit you won't have access to your answers

DON'T FORGET TO CLICK THE 'SUBMIT FOR GRADING' BUTTON WHEN YOU'RE DONE, OR ELSE YOUR ASSIGNMENT WON'T GO THROUGH AND YOU WILL RECEIVE NO CREDIT!

Note if you are the type of person who likes to change their answers then you should save your answers in a document. Once you hit submit for grading thats it. If you need to change an answer after that then you will need to redo the entire quiz.

Most of the questions are set up in such a way so that T-Square can grade them automatically. Please make sure that you follow all of the rules or else you will lose points.

1) Wherever possible, always the actual number required instead of a formula or English words.

i.e. What is three multiplied by itself 4 times?
GOOD: 81
BAD: 3*3*3*3, Eighty-one

2) If a formula is required, input it without any spaces, and don't use English.
i.e. What is three multiplied by X?
GOOD: 3*X , X*3
BAD: 3 * X, 3 times X

3) When asked for a binary number, input it without any spaces, and use the exact number of bits requested in the problem
i.e. Using 8 bits, what is the number '12' in 2's complement binary?
GOOD: 00001100
BAD: 0000000000001100, 1100, 0x0C, 0xC, 0000 1100

i.e. What is an encoding for 'Not a Number' in IEEE Floating Point?
GOOD: 01111111111011011001100111001011
(Note that this is simply putting the sign bit (0), the exponent (11111111), and the mantissa (11011011001100111001011) back-to-back with no spaces.)
BAD: 0 11111111 11011011001100111001011, 0 11111111 1101 1011 0011 0011 1001 011

Part 2

Now for some coding complete the methods given in the files HW2BitVector.java and HW2Operations.java. All of the methods have javadocs indicating what they should do so make sure you read them.

Evaluation note: We will be grading this on a system with JDK 6 installed. Also note that if we can't compile your code you will receive a zero.

Each of these files have limitations on what you can and can't use to code them so please look at the comments at the top of each file.

If you use anything that is banned you will get no credit for that function. The point of this is to play with bit operations and learn how they work. You will not have all of these fancy things when we get to C.

Coding Guidelines for HW2.

1. All bitmasks **MUST** be written in hexadecimal. This is the convention for masks and makes it very easy for the programmer to understand what you are doing. If you write a mask in any other base you will lose points.
    - GOOD: (num & **0xFFE0**) >> 5 | (num & **0xF01F**) << 2 ^ **0x1**
    - BAD: (num & 65504) >> 5 | (num & 61471) << 2 ^ 1
    - Notice with the GOOD you can more easily tell what the line of code is doing.
    - Note the bolded things are bitmasks!
2. You are required to comment and test your code failure to do so will result in point deductions.

Hints

Remember that all numbers are stored in your computer as binary. So when you perform these operations the computer does it for you. All you need to do is tell the computer how you are representing your numbers and how to interpret them.

You can specify your number in decimal octal and hexadecimal example.
System.out.println(16);
System.out.println(020);
System.out.println(0x10);

Are three ways of representing 16 in these bases.

You can also tell Java to print out your number in different bases using a method called printf.
printf is the GRANDDADDY of all printing functions and when we get to C you will have to fall in love with it.

printf takes a variable number of arguments the first of which is a format string.
and then parameters. What gets printed is controlled by the format string.

Example

```
System.out.printf("In decimal: %d", 16);
System.out.printf("In octal: %o", 16);
System.out.printf("In hexadecimal: %x", 16);
```

The %d,%o,%x get replaced by the parameter passed in.
printf does not support printing the number out in binary.

For more information about printf read http://en.wikipedia.org/wiki/Printf
(temporary link: http://www.cplusplus.com/reference/clibrary/cstdio/printf/)

Evaluation

1. Correctness of your answers to quiz portion
2. Correctness of the output of your code
3. Implementation of code without banned operations
4. We will be running test cases of our own. Note that we will be using Java 6 to grade your code.

Deliverables

A submitted version of the quiz HW2 under Test and Quizzes.
The file HW2BitVector.java

The file HW2Operations.java