CPSC 229, Fall 2015 — Second Test Info

The second test for this course will be given in class on Wednesday, October 21. It covers everything that we have done since the first test, starting with mathematical induction (Section 1.8) and ending with regular expressions (Section 3.2). Note that we skipped large parts of Chapter 2; we covered all of Sections 2.1, 2.3, and 2.6, plus approximately the first halves of 2.2 and 2.4.

As usual, the test will include some "essay-type" questions that ask you to define something, or discuss something, or explain something, and so on. Other than that, you can expect most of the questions to be similar to problems that have been given on the homework. This might include some proofs, including possibly a simple proof by mathematical induction. There might be some questions related to the programming assignment about Java bitwise operations and using integers in Java to represent sets.

Note: Class on Friday, October 23 will be in Rosenberg 009.

Here are some terms and ideas that you should be familiar with for the test:

the principle of mathematical induction
proof by mathematical induction
why mathematical induction works
summation notation, for example: $\sum_{k=1}^{n} a_k$
sets
set notations: $\{a,b,c\}$, $\{1,2,3,\dots\}$, $\{x \mid P(x)\}$, $\{x \in A \mid P(x)\}$
the empty set, $\emptyset$ or $\{\,\}$
equality of sets: $A = B$ iff they contain the same elements
element of a set: $a \in A$
sets can contain other sets as elements
subset: $A \subseteq B$
$A = B$ if and only if both $A \subseteq B$ and $B \subseteq A$
union, intersection, and set difference: $A \cup B$, $A \cap B$, $A \smallsetminus B$
definition of set operations in terms of logical operators
disjoint sets ($A \cap B = \emptyset$)
power set of a set: $\mathcal{P}(A)$
universal set
complement of a set (in a universal set): $\overline{A}$
DeMorgan's Laws for sets
bitwise operations in Java: `&`, `|`, `~`
using an n-bit integer to represent subsets of $\{0,1,2,\dots,n-1\}$
`&`, `|`, and `~` as set operations (intersection, union, complement)
the shift operators `<<`, `>>`, and `>>>`
using "`1 << n`" to represent a set with one element, $\{n\}$

hexadecimal numbers
ordered pair: (a, b)
cross product of sets: $A \times B$
function $f \colon A \to B$
domain and range of a function f
one-to-one correspondence
cardinality of a finite set: $|A|$
finite set (in one-to one correspondence with one of the sets $N_0, N_1, N_2, \dots$)
infinite set (not finite)
countably infinite set (in one-to-one correspondence with $\mathbb{N}$)
a set is countably infinite iff its elements can be placed into an infinite list
countable set (finite or countably infinite)
uncountable set (that is, uncountably infinite)
examples of countably infinite and uncountably infinite sets
the union of two countably infinite sets is countably infinite
if X is uncountable and A is a countable subset, then $X \smallsetminus A$ is uncountable
for any set A, there is no one-to-one correspondence between A and $\mathcal{P}(A)$
the power set of a countably infinite set is uncountable
alphabet (finite, non-empty set of "symbols")
string over an alphabet Σ
length of a string, $|x|$
empty string, ε
concatenation of strings, xy or $x \cdot y$
reverse of a string, x^R
x^n, for a string x and a natural number n
$n_\sigma(x)$, the number of occurrences of a symbol σ in a string x
the set of all possible strings over Σ, denoted Σ^*
language over an alphabet Σ (a subset of Σ^*)
a language over Σ is an element of $\mathcal{P}\big(\Sigma^*\big)$
the set of strings over Σ is countable; the set of languages over Σ is uncountable
operations on languages
union, intersection, set difference, and complement applied to languages
concatenation of two languages: LM
L^n, for a language L and a natural number n
Kleene star of a language: L^*
regular experssion over an alphabet Σ
the regular expression operators: `*`, `|`, and concatenation
regular language; the language $L(r)$ generated by a regular expression r
finding the regular expression for a given language
finding the language generated by a given regular expression
not every language is regular