Name: ____

CS 620 Fall 2010 at ISU, Exam 3 SAMPLE

Prepared by assistant professor Jeff Kinne on November 30, 2010. You have until 4:00pm to finish the exam. The exam itself and blank paper I provide are all you will have to use (no computer, textbook, notes, cellphone, calculator, etc.). I have put point values on the problems so the total adds up to 28. I **WILL** collect the exams at 4:00pm, so budget your time so you have time to answer each question.

Problem 1 (3 points) Show the following function is *not* a one-way function.
Addition: $f(x, y) = x + y$.

Problem 2 (3 points) Show the following is *not* a one-way function.
Factoring: $f(n) = (p_1, p_2, ..., p_k)$ where these are the prime factors of n in order from smallest to largest and there may be duplicates if needed.

Problem 3 (5 points) Let f be any function that is computable in polynomial time. Show that if f is a one-way function, then P $\neq$ NP. Let f be any function that is computable in polynomial time. Show that if f is a one-way function, then P $\neq$ NP.

Problem 4 (5 points) Let G be a one-one function that maps n bits to $n+1$ bits. Define f, a function from $n+1$ bits to 1 bit such that $f(y) = 1$ iff y is in the range of G (namely, $f(y) = 1$ iff there is an x such that $G(x) = y$). Show that if you can compute f correctly on $\frac{1}{2} + 1/\text{poly}$ fraction of inputs, then you can distinguish the output of G from uniform with 1/poly advantage.

In other words, if G is pseudorandom, then f is hard to compute. It can also be shown that PRGs imply one-way functions.

Problem 5 (5 points) Explain why every language in BPP has a zero-knowledge proof system. In particular, what is the prover, what is the verifier, and what is the simulator?

Problem 6 (3 points) Let f be a poly-time 1-1 length-preserving function from n bits to n bits, and let b be a poly-time function from n bits to 1 bit. Show that if b is hard-core for f, then f is a one-way function.

Problem 7 (5 Points) Let f be a one-way function, and let f' be a function from $k \cdot n$ bits to $k \cdot n$ bits defined by $f'(x_1, x_2, ..., x_k) = f(x_1), f(x_2), ..., f(x_k)$. If we use the naive strategy of trying to invert f' by using some poly-time algorithm to independently invert f on each of $f(x_1), ..., f(x_k)$, then give an upper bound on the probability of success of this strategy.