Instructions:

1) This test is 4 pages in length.

2) You have 75 minutes to complete and turn in this test.

3) Short-answer questions include a guideline for how many sentences to write. Respond in complete English sentences.

4) This test is closed books, notes, papers, friends, neighbors, etc.

5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.

6) Write and sign the following: “I pledge my Honor that I have not cheated, and will not cheat, on this test.”

_______________________________________________________________________

Signed:  ______________________________________________
1. [5 points]  
What are first-class functions? [1-2 sentences] 

2. [10 points]  
a) Provide an example of an ML program that violates the value restriction.  
b) Rewrite your example from Part (a) into an equivalent program that does not violate the value restriction.  

3. [25 points]  
a) Implement a function filter that takes (in curried form) a function F and a list L of triples. Function F must take a triple and return a bool. Function filter returns a list containing only those triples in L for which F returns true. Use ML syntax in your implementation (including pattern matching, anonymous variables, and as-bindings when appropriate). Do not call any built-in higher-order functions (like map, foldl, or foldr) in your implementation.  
b) What type does filter have?
4. [20 points]
Consider the following function F.

   fun F g s = foldl (fn(x,y)=>(g(x) andalso y)) true s;

a) What is the type of F?

b) Succinctly summarize what function F does. (1 sentence)

c) What is the type of the expression $F (fn x=>x<5) [2,4,~6,~8,~3,5,~6,~10]$?

d) To what value does $F (fn x=>x<5) [2,4,~6,~8,~3,5,~6,~10]$ evaluate?

e) What is the type of the expression $F (fn x=>x)$?

f) Implement the simplest possible function that is equivalent to F but that does not use a built-in function like foldl. Use ML syntax (including pattern matching, anonymous variables, and as-bindings when appropriate).
5. [40 points]
   a) Define inference rules for greater-than and less-than judgments over natural numbers. The judgment forms are: \( N_1 > N_2 \) and \( N_1 < N_2 \). Recall that \( N_1 \) and \( N_2 \), being natural numbers, adhere to our definition of natural numbers (\( N_{\text{nat}} \)), as discussed in class.

   b) Using your definitions of greater-than and less-than, prove that for all natural numbers \( N_1 \) and \( N_2 \): \((N_1 > N_2) \) if and only if \((N_2 < N_1)\).

   c) Prove that your less-than operator is transitive, i.e., for all natural numbers \( N_1, N_2, \) and \( N_3 \): if \( N_1 < N_2 \) and \( N_2 < N_3 \) then \( N_1 < N_3 \).