Name:____ Grade:_____<-- instructor use

1. When is a (propositional) wff a *tautology*? When it is true for all values of its variables

2. When we say a set of inference rules is *sound*, what do we mean? If we apply the rules to a set of axioms, we only end up with things entailed by those axioms

3. What is a *predicate*? A function whose value is Boolean

   Give an example of a predicate application with no free variables Example: contains(3, {4, 5, 6})

   with one or more free variables Example: contains(n, {4, 5, 6})

4. When is a first-order wff a *sentence* (statement)? When it has no free variables

5. Give an example of a model for $\exists x\ (\forall y\ (xy = 0))$ Integers, with standard definitions of 0 and <

6. From { ∀t(p(t)→q(t)), ∀t(q(t)→r(t)), ¬r(C)}, prove ¬p(C). Give reasons for your steps. (Continue on back)

1. ∀t(p(t)→q(t))	given
2. p(C)→q(C))	1, universal instantiation
3. ∀t(q(t)→r(t))	given
4. q(C)→r(C))	1, universal instantiation
5. p(C)→r(C))	2, 4, syllogism
6. ¬r(C)}	premise
7. ¬p(C)}	modus tollens

7. Consider the set of ordered pairs of non-negative integers. Working with another student, define a relation on this set that is a total ordering.

   Call the relation #. Use lexicographic order.
   (a, b) # (c d) iff either $a \leq c$ or ($a = c$ and $b \leq d$)

   A student suggested (a, b) # (c d) iff $\text{sqrt}(a^2+b^2) \leq \text{sqrt}(c^2+d^2)$)

8. Working with another student, define a relation on the positive rational numbers that is a total ordering. Note that the ordering is on numbers, not just on representations of the numbers.

   This one turned out to be trivial. The standard ≤ works fine.

9. Working with another student, define a well-ordered relation on the rational numbers r with $0 < r < 1$.
   Hint: Think diagonal.

   Given a positive rational number r, let f (r) be the **reduced** fraction that represents r. Let n(r) be the numerator of f (r), and d(r), be the denominator of f (r). Then the ordering can again be lexicographic:
   r # s if either $d(r) < d(s)$ or ($d(r) = d(s)$) and $n(r) < n(s)$).

   Note that without the restriction to representations by reduced fractions, this is not even a **partial** ordering. If we say that a fraction is simply a numerator and a denominator with no restrictions, we get 1/2 # 2/3 and 2/3 # 2/4. But 1/2 and 2/4, while they are different "fractions", represent the same rational number.

   This order is a well-order, which begins
   1/2, 1/3, 2/3, 1/4, 3/4, 1/5, 2/5, 3/5, 4/5, 1/6, 5/6, 1/7, ...

10. Working with another student, use (strong) induction to prove by that for any natural number n, n(n+1)(n+2) is divisible by 6.

    Base case: n=0: (0)(1)(2) = 0(6).

    Induction step: Show $\forall j > 2$ (($\forall k$ ($0 \leq k < j \rightarrow$ k(k+1)(k+2) divisible by 6) → j(j+1)(j+2) divisible by 6)
    So the induction hypothesis is ($\forall k$ ($0 \leq k < j \rightarrow$ k(k+1)(k+2) divisible by 6) and what we need to deduce from it is j(j+1)(j+2) divisible by 6.

    So here we go! Let j>3. The particular k we want to use is j-1 (so it turns out that ordinary induction would have worked). By the induction hypothesis, we have (k)(k+1)(k+2) = 6m for some m. We want to show that (k+1)(k+2)(k+3) is a multiple of 6. We can write this as this as k(k+1)(k+2) + 3(k+1)(k+2) = 6m + 3(k+1)(k+2). One of k + 1 and k + 2 must be odd, and the other must be even, so their product must be even. Thus 3(k+1)(k+2) is divisible by 6, and 6m is divisible by 6, so their sum is divisible by 6.

11. Tell your instructor about anything from today's session (or from the course so far) that you found confusing or still have a question about. If none, please write "None". Continue on the back if needed. Must have some answer

Additional material: Detailed proof of in-class example.

- From Weiss, Data Structures and Problem Solving with Java, Section 7.3.4
- Consider this function to recursively calculate Fibonacci numbers:
  $F_0=0$ $F_1=1$ $F_n = F_{n-1}+F_{n-2}$ if $n\geq 2$.
  - **def fib(n):**
    if n <= 1:
    return n
    return fib(n-1) + fib(n-2)
- Let C_N be the number of calls to fib during the computation of fib(N).
- It's easy to see that $C_0=C_1=1$,
  and if $N \geq 2$, $C_N = C_{N-1} + C_{N-2} + 1$.
- **Prove that** for $N \geq 3$, $C_N = F_{N+2} + F_{N-1} - 1$.

Base cases:
If N=3, then $C_N = 5$. $F_5 + F_2 - 1 = 5 + 1 - 1 = 5$. ☑
If N=4, then $C_N = 9$. $F_6 + F_3 - 1 = 8 + 2 - 1 = 9$. ☑

Induction step: Show $\forall j > 4$ $((\forall k\ (3 \leq k < j \rightarrow C_k = F_{k+2} + F_{k-1} - 1) \rightarrow C_j = F_{j+2} + F_{j-1} - 1)$

The two particular values for k that we use are j-1 and j-2.
Thus $C_{j-2} = F_j + F_{j-3} - 1$ and $C_{j-j} = F_{j+1} + F_{j-2} - 1$

Now we can prove the conclusion:

$C_j = C_{j-1} + C_{j-2} + 1$ (from the next-to-last bullet in the statement of the problem)
$= (F_{j+1} + F_{j-2} - 1) + (F_j + F_{j-3} - 1) + 1$ (induction assumption)
$= (F_{j+1} + F_j) + (F_{j-2} + F_{j-3}) + 1 - 1 - 1$ (commutative and associative laws)
$= F_{j+2} + F_{j-1} - 1$ (def of Fibonacci numbers)