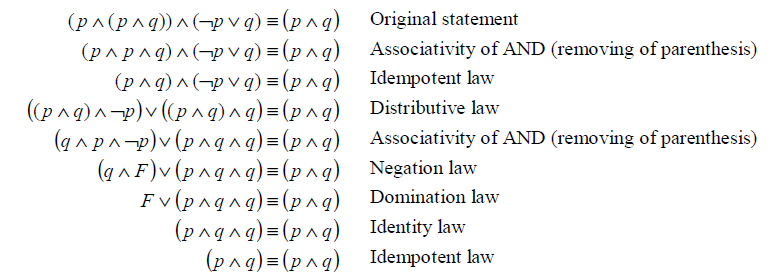
***Discrete Math1*. Handwriting Assignment #1 Solution**

1. Give the converse, the contrapositive, and the inverse of the statement “If it rains today, then I will drive to work.”  
   **Solution)  
   The converse is “If I drive to work today, then it will rain.”  
   The contrapositive is “If I do not drive to work today, then it will not rain.”  
   The inverse is “If it does not rain today, then I will not drive to work.”**
2. Given the Boolean proposition p↔q, write an equivalent compound proposition using only the operators ¬, ∧, and ∨.  
   **Solution) ￢*p* ∨*q*∧￢*q* ∨*p***
3. Show that ￢*p →(q→r*) and *q→*(*p*∨*r*) are logically equivalent using the laws of logical equivalences. Be sure to cite each law whenever used.  
   **Solution)**
4. Prove that ( *p* ∧ ( *p* ∧ *q*)) ∧ (￢*p* ∨ *q*) is equivalent to (*p*∧*q*) using logical equivalences. You must clearly label each step of the logical equivalence.  
   Solution)  
   
5. Prove that NOR is functionally complete.  
   Solution)   
   **To show that NOR is functionally complete, we must show that NOT, AND, OR can be expressed as only NOR.  
   NOT: : p**↓ **p= ￢(p**∨**p) =￢p  
   AND: pq=￢(￢p**∨**￢q) =￢p**↓**￢q=( p**↓ **p)** ↓ **(q**↓ **q)  
   OR: p**∨**q=￢(￢(p**∨**q)= ￢(p**↓**q)=( p**↓**q)** ↓ **(p**↓**q)  
   Hence, NOR is functionally complete**
6. Whenever expressions are connected together using the logical operators &&(AND) or || (OR), only as many expressions as are needed to determine the overall logical value will be evaluated. This is known as *short-circuit* behavior. For example, if two expressions are connected with && and the first expression is false, then it is guaranteed that the second expression will not be evaluated. The same is true of expressions connected with || where the first expression is true.  
   Write the output of the following C code segment.

int i, j;

i = 2 && (j = 2); /\* what is printed \*/

printf(“%d %d\n”, i, j);

(i = 0) && (j = 3);

printf(“%d %d\n”, i, j);

i = 0 || (j = 4);

printf(“%d %d\n”, i, j);

(i = 2) || (j = 5);

printf(“%d %d\n”, i, j);

**Solution  
 1 2**

**0 2**

**1 4**

**2 4**