**Name**

**Advanced Programming in Java**

**Lab Exercise 11/18/2020**

**Boolean Algebra and DeMorgan’s Theorem**

Assume in the following *boolean* expressions that *a*, *b*, *c*, and *d* are *boolean* quantities.

1. Show the algebraic simplification of ab(!c) + !ab(!c) to b(!c).

2. Show the algebraic simplification of a(!b)c + abc + !abc + !ab(!c) + a(!b) to

!ba + !ab + ac

3. Show the algebraic simplification of !(!a \* !c) + !b + !(ad) to *true*.

4. Show the algebraic modification of !( (a + bc) (a + c) ) to !a(!b + !c) + !a(!c).

5. Express ab(!c) + !ab(!c) using proper Java syntax (&&, ||, and !).

6. Express a(!b)c + abc + !abc + !ab(!c) + a(!b) using proper Java syntax (&&, ||, and !).

7. Express !(!a \* !c) + !b + !(ad) using proper Java syntax (&&, ||, and !).

8. Simplify x + (x \* y) where *x* and *y* are *boolean* variables.

9. Express !( (a + bc) (a + c) ) using proper Java syntax (&&, ||, and !).

10. Express !(a && b && c) using **OR**s instead of **AND**s.

11. Express (a && b && c) using **OR**s instead of **AND**s.

12. Illustrate the equivalence of !(a \* b) and !a + !b using truth tables.

13. Which of the following is the equivalent of ( (p > 3) || (q < b) ) ?

a. !( !(p>3) && !(q<=b) )

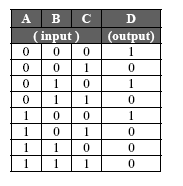
b. !( (p <= 3) && (q >= b) )

c. !( (p > 3) || (q < b) )

d. !( !(p <= 3) || (q >= b) )

e. More than one of these

14. Write a Boolean expression that produces the following truth table:



**Programming Assignments**

1. Neural networks are non-linear statistical models that “learn” from experience and are used in pattern recognition systems. Neural network algorithms need to calculate the “sigmoid” function which can be described using the following equation:



Write a class Sigmoid that tabulates the sigmoid function and provides a method that returns it’s value from a table.

1. Your constructor

public Sigmoid(double step)

tabulates the values of the sigmoid for -10 ≤ x ≤ 10 with the given step and places the values into a private field double myValues[]. For example, if the step was 0.01, then there would be 2001 elements. You will need to figure out an good way to find the number of elements for any given step.

1. Write a method

public double getValue(double x)

that fetches and returns the sigmoid value -10 ≤ x ≤ 10 from the table. The method should return 0 if x < -10 and 1 if x > 10.

1. Write an application to test your application.
2. Write an Email class that is used to store user name, domain and email address. You should have the following private fields:

myUserName

myDomain

myEmail

Provide public set and get methods as well as constructors. Your constructor should not construct the email object unless the email is valid. Provide the following methods:

public boolean validEmail() //used by constructor

public static boolean validEmail(Email em) //used test any email for validity