EECE 495 (Summer 2012)
Individual Programming Assignment (Lab Assignment 1)
Due Date: June 25 (Beginning of class)

1. **Write a Java Program that accepts a number from the user, computes its factorial and returns the sum of digits in the factorial.**
   (If user enters 10, then output of the program should reflect sum of digits in 10!)

   - Program must use **at least one example of try and catch to handle errors** (for example, if user inputs something other than an integer) (5 points will be deducted from "results are correct" section, explained in grading section at the end of document if they do not)

2. **Design and implement *login form* functionality using *Swing classes*, subject to following basic requirements:**

   2.a) The user interface should have controls for accepting username and password, with appropriate labels and two buttons, one for 'Submit' and other 'Reset'.
   2.b) Password textfield should not display the user input. Use asterisks to mask the user input.
   2.c) Hitting 'Reset' button should clear the textfields for username and password.
   2.d) Hitting 'Submit' should validate the entered values on the form against hard coded values in your code. Validate the user input against following values:
   username: eeceSummer12
   password: gobearcats!
   2.e) Allow three attempts to the user. For invalid username/password during first attempts, display authentication failure message in a pop-up message box and reset textfields to allow user to try again.
   2.f) After three invalid attempts, display failure message to user and exit the application.

- Both of your programs should be:
  a) WELL-STRUCTURED
  b) WELL-DOCUMENTED (INCLUDING APPROPRIATE VARIABLE,CONSTANT, AND METHOD NAMES)

NOTE: What does "well-documented" mean?
--program should tell who wrote it and when
--every variable and function name should be meaningful
--every class and method should have a sentence at the beginning explaining its purpose
--a programmer who might need to modify this code later should be able to get a general understanding of it from reading the comments

--it is not being written for a beginning programmer, so you needn't comment EVERY LINE

c) WELL-TESTED (include comments about what test cases you used)
d) LABELED with your name and the date completed

PROGRAM GRADING (general rules for quarter):

20%--compiles correctly
20%--executes (on designated machine), ends gracefully
30%--results are correct
10%--well-structured java code, uses language appropriately
10%--well-documented, including variable and method names;
10%--attractive user interface, user-friendly