CMPS4143: Contemporary Programming Languages
Programming Assignment 5 (75 points) **Due: Wednesday, Oct. 21, 2020**

PURPOSE: To create a library (dll) of critters; to instantiate objects of classes in an inheritance hierarchy and polymorphically output each object's attributes; to use static variables; to use a labels, text boxes and buttons; to use the switch statement; to use exception handling; to use a data structure to hold references to objects. *Remember you will be graded on applying the concepts learned in class.*

PROBLEM: **Plants, Majestic Plants, Deadly Mimics, Flies, Oh My!**
Design an inheritance hierarchy for class the above actors. Use Actor as the ***base*** class of the hierarchy. (Use private data.) Write a program that randomly generates data for a "Game of Life" automata. The program should

- prompts the user for the number of each type of actor (the Majestic plant, the Deadly Mimic, and Flies)
- randomly generate positions in the grid (making sure no 2 objects are in the same location)
- displays a **picture** of the actor on the grid (DO NOT DRAW THIS. Find images online.)
- prompt the user for the number of generations
- run a simulation of the Game of Life, allowing the user to view each generation at the leisure or just showing the final result of apply the rules in the game of life as described below
- performs exception handling for negative (or illegal) values (and then allows user to try again)

The rules in the game of life are as follows: Majestics are Plants, that when pollinated, sprout and grow for 2 more generations. Deadly mimics do all that and when they eat, they grow larger for 2 more generations. BUT if they don't eat for 5 generations, they die. Flies eat Majestics and live another day. They can go 5 generations without eating. However, if a poor unsuspecting fly encounters a Deadly Mimic, they get eaten and cease to exist.

INPUT: Sample INPUT values on next page. User should be able to
- Grid size
- Number of each Majestic, Deadly Mimic and Fly
- Number of generations to run the Game of Life.

OUTPUT: Output is all done on a form. The form should display
- Buttons to select options
- A Grid of images
- Input Generation Number and Current Generation
- End of the game, output number of remaining objects of each type.

TURN IN (upload) all materials:
- Documented Source code
- Screen dump(s) of image(s) when running
- Zipped application